



GPL162002A/162003A Programming Guide

V1.0 – Dec. 20, 2006



Important Notice

Generalplus Technology reserves the right to change this documentation without prior notice. Information provided by Generalplus Technology is believed to be accurate and reliable. However, Generalplus Technology makes no warranty for any errors which may appear in this document. Contact Generalplus Technology to obtain the latest version of device specifications before placing your order. No responsibility is assumed by Generalplus Technology for any infringement of patent or other rights of third parties which may result from its use. In addition, Generalplus products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Generalplus.

For Generalplus Confidential Use Only
普格科技股份有限公司

Table of Content

	<u>PAGE</u>
1 CONFIRMATION SHEET	10
2 INTRODUCTION	11
2.1 General Description	11
2.2 Significant Features	11
2.3 Applications	12
2.4 The Differences between GPL162002A and GPL162003A	12
3 SYSTEM CONTROL	13
3.1 Introduction	13
3.2 Device Identification	14
3.3 Reset Control	14
3.4 Clock Generation	15
3.5 System Reliability Control	18
3.6 Operation Mode Control	19
3.7 Special Note	23
4 MEMORY	24
4.1 Introduction	24
4.2 FEATURE	24
4.3 Memory Mappings	24
4.4 Memory Access Pin Configuration	29
4.5 Control Registers	29
4.6 Bank Switch Control	38
4.7 Vectors	40
4.8 Stack Location	40
4.9 Chip Select (Project Setting) on IDE	40
4.10 Program Examples	41
5 I/O PORTS	42
5.1 Available Ports	42
5.2 General Purpose I/Os Configuration	47
5.3 General Purpose I/Os Function Table	47
5.4 Control Register	48
5.5 I/O structure (diagrams)	53
5.6 Special Notes	54
5.7 Program Examples	54
6 INTERRUPT	55

6.1	Introduction	55
6.2	Peripheral Interrupt Arrangement.....	57
6.3	Control Registers	60
6.4	Program Examples.....	67
7	TIMER/COUNTER	68
7.1	Timer Introduction	68
7.2	Timer Structure and Clock Source	68
7.3	Control Registers	73
7.4	Program Examples.....	79
8	TIMEBASE.....	80
8.1	Introduction	80
8.2	Timebase structure and clock source.....	80
8.3	Control Registers	80
8.4	Program Examples.....	84
9	REAL TIME CLOCK (RTC).....	85
9.1	Introduction	85
9.2	RTC Structure and clock source.....	85
9.3	Control Registers	85
9.4	Program Examples.....	91
10	AUDIO OUTPUT	93
10.1	DAC.....	93
10.2	DAC Operation.....	93
10.3	Block Diagram.....	94
10.4	Speech Mode.....	94
10.5	IIS Mode	94
10.6	Control Registers.....	96
10.7	Mute level.....	108
10.8	Ramp Up and Ramp Down.....	108
10.9	3D/EQ/AC parameter reference setup.....	109
10.9.1	EQ/AC Control Method.....	109
10.9.2	3D Control Method	110
10.10	Program Examples	110
11	STN LCD.....	113
11.1	Introduction.....	113
11.2	LCD Control Pin Configuration	113
11.3	LCD Buffer.....	114
11.4	LCD Palette	115

11.5	Control Registers	117
11.6	Operation during Wait/Halt/Standby & Wakeup Procedure	126
11.7	LCD Image Resource File Tooling	127
11.8	Program Examples	128
12	TFT LCD	131
12.1	Introduction	131
12.2	TFT Interface Signal	131
12.3	TFT LCD Buffer	132
12.4	Sub Frame Display (PIP)	132
12.5	Control Register	133
12.6	Operation during Wait/Halt/Standby & Wakeup Procedure	152
12.7	Programming Example	152
13	UART/IRDA INTERFACE	154
13.1	Introduction	154
13.2	Structure and Block Diagram	154
13.3	UART / IrDA SIR Frame Scheme	155
13.4	UART/IrDA Control Pin Configuration	156
13.5	Control registers	156
13.6	Program Examples	167
14	SERIAL PERIPHERAL INTERFACE (SPI)	168
14.1	Introduction	168
14.2	Structure	168
14.3	SPI Control Pin Configuration	169
14.4	Master Mode	169
14.5	Slave Mode	169
14.6	Consecutive Bytes Transfer	172
14.7	Control Registers	172
14.8	Program Examples	178
15	USB INTERFACE	179
15.1	Introduction	179
15.2	USB Device	180
15.3	USB Mini-Host	180
15.4	Serial Interface Control Pin Configuration	180
15.5	Control Registers	180
15.6	USB Device Register Definition	182
15.7	USB Host Register Definition	203
16	ANALOG INPUT (TOUCH PANEL / VOICE RECORDER)	214

16.1	Introduction.....	214
16.2	SAR ADC Control	214
16.3	Touch Panel Interface.....	215
16.4	Voice Recorder (HQADC operates mode).....	220
16.5	Analog Input Control Pin Configuration	221
16.6	Control Registers.....	222
16.7	Program Example.....	232
17	NAND FLASH INTERFACE.....	235
17.1	Introduction.....	235
17.2	NAND Flash Control Pin Configuration.....	235
17.3	Control Register.....	235
17.4	NAND Flash ECC & Checksum.....	239
17.5	Special Note	246
17.6	Program Example.....	247
18	I2C CONTROLLER.....	249
18.1	Introduction.....	249
18.2	I2C Bus Protocol.....	249
18.2.1	Start / Stop Generation	249
18.2.2	Data Transfer Format	250
18.2.3	Acknowledgement Signal Transmission	251
18.2.4	Read / Write Operation.....	252
18.2.5	Bus Arbitration Produres	252
18.2.6	Bus Arbitration Produres	252
18.3	Firmware Flow Chart.....	253
18.3.1	Master Transmit Mode.....	253
18.3.2	Master Receive Mode	254
18.3.3	Slave Transmit Mode.....	255
18.3.4	Slave Receive Mode	256
18.4	I2C Bus Control Pin Configuration.....	257
18.5	I2C Bus Control Register	257
18.6	Example Program.....	261
19	DMA AND BRIDGE CONTROLLER.....	263
19.1	Introduction.....	263
19.2	Block Diagram	264
19.3	Operation Mode.....	264
19.3.1	Single Mode	264
19.3.2	Demand Mode.....	265

19.4	Double Buffer Mode.....	265
19.5	Byte Mode Operation.....	266
19.6	Control Register.....	267
19.7	Program Examples.....	281
20	SD AND MMC MEMORY/IO CARD CONTROLLER	282
20.1	Introduction.....	282
20.2	Block Diagram	282
20.3	Command Line Control.....	283
20.4	Data Line Control.....	283
20.5	Card Insertion Detection	284
20.6	Multi-Block Read/Write	284
20.7	SD/MMC Control Pin Configuration	284
20.8	Control Register.....	285
20.9	Example Program.....	292
21	KEY SCAN CONTROLLER	294
21.1	Introduction.....	294
21.2	Key Scan Function	294
21.3	Key Scan Application Circuit.....	294
21.4	Sample Time Configuration.....	296
21.5	Auto / Manual Sample Mode.....	297
21.6	Automatically Detect Key Process.....	297
21.7	Key Scan Control Pin Configuration	298
21.8	Control Register.....	298
21.9	Example Program.....	303
22	MISCELLANEOUS	305
22.1	Introduction.....	305
22.2	Specified Register.....	305
22.3	Program example	306
23	E-FUSE OPTION	307
23.1	Introduction.....	307
23.2	Specified Register.....	307
24	LINK YOUR PROGRAM	309
24.1	Resource File Alignment by Link Script File	309
25	APPENDIX.....	311
25.1	Normally used abbreviation list.....	311
25.2	Control Register Mapping List (by function).....	312
25.3	Crystal Usage Guide	371

25.4	Development (Emulation) Board Configuration	372
25.4.1	Power Adjustment & Selection	375
25.4.2	Reset.....	375
25.4.3	ICE	375
25.4.4	12MHz Crystal.....	375
25.4.5	CPU Boot mode	376
25.4.6	Memory	376
25.4.7	Audio Output	376
25.4.8	Analog Input.....	376
25.4.9	Others Input and Output.....	378
25.5	Development (Emulation) Board Schematic.....	381
25.6	CPU Performance Downgrade Issue.....	393
25.6.1	LCD Display and System Performance	393
25.7	Audio Output Components Selection Guide	394
25.8	32768 Crystal and PLL Power-on Stable Time	395
25.9	Reset Type	395
25.10	Important Note for the Setup of Memory Access Time	396
25.11	Project Setting on IDE	397

Revision History

Revision	Date	By	Remark
1.0	12/20/2006	Jacky Lin	1. Add GPL162003 body. 2. Correct relationship table between BM2 and USB function. Please see section 3.4 for detail and see Revision table1. 3. Correct 0x7815 congtrrol register description. For detail, please see system control chapter. 4. Correct 0x7D14, 0x7D1F, 0x7D2A, 0x7D35 control register description. For detail, please see TFT LCD chapter.
0.1	06/15/2006	Jacky Lin	First edition

Revision 1.0
Revision table1

BM2	USB Function	Slow PLL (32768Hz -> 12MHz)	Fast PLL (12MHz -> 96MHz)
0	Not Active	Active	Active (After enter Fast mode)
1	Available when Fast PLL is on	Not Active	Active (After enter Fast mode)

1 Confirmation sheet

The confirmation sheet, as a requisite document before placing orders, contains useful information and checklist that help to avoid mistakes during development. Due to the updates of the confirmation sheet, please download a newest confirmation sheet from <http://www.generalplus.com/>.

For Generalplus Confidential Use Only
普格科技股份有限公司

2 Introduction

2.1 General Description

The GPL162002A/162003A, a 16-bit architecture LCD controller product, carries the Sunplus newest 16-bit microprocessor, called $\mu'nSP^{\circledR}$ (pronounced as *micro-n-SP*). This high processing speed assures the $\mu'nSP^{\circledR}$ is capable of handling complex digital signal processes easily and rapidly. Therefore, the GPL162002A/162003A is applicable to the areas of digital sound process, voice recognition and learning auxiliary product. The memory capacity includes a 30K-word working SRAM and 128K-word ROM. The LCD controller is able to support maximum 320x320 dots 16 gray-level or 4096 CSTN or 640X480 dots 65536 color TFT displays. Other features include 64 programmable multi-functional I/Os, MP3 decode accelerator, six 16-bit timers/counters, 32768Hz Real Time Clock, Low Voltage Reset, 12-bit ADC for touch panel and general-purpose application, 16-bit ADC/DAC for voice playing/recording, supports SD memory card, USB 1.1version, and plus many others.

The control registers of MP3 decode accelerator will not be described in this programming guide. Please contact Generalplus for details of MP3 decoding accelerator and decoding process.

2.2 Significant Features

- Sunplus16-bit CPU ($\mu'nSP^{\circledR}$) maximum 96MHz @ 2.7V ~ 3.6V.
- Dual Clocks System (Phase Lock Loop and 32768 Crystal).
- Flexible Operations: Wait//Halt/Sleep for power management.
- Address extensible to 80M words.
- Built-in Internal 30K-word SRAM.
- STN LCD controller supporting up to 320x320 dots 16 gray-level or 4096 color level display.
- TFT LCD controller supporting up to 640X480 dots 65536 color level display.
- Two Channels 16-bit DAC audio outputs.
- MP3 decoding accelerator.
- 7-band programmable equalizer.
- 3D Surround processor.
- Six channels 12-bit ADC, two channels are dedicated to touch panel.
- 16-bit ADC for stereo microphone/line-in/FM record.
- One UART & One IrDA with 8-byte transmit and receive FIFOs (queues).
- Five chip select pins to access external ROM, SRAM, and NOR & NAND-Flash memories.
- Six 16-bit re-loadable timers/counters and two of them support capture, comparison, and PWM functions.
- One SPI, Serial Peripheral Interface.

- Real Time Clock (RTC) supports auto-update to hour and an alarm comparison register.
- Built-in 2.5V low voltage reset.
- Embedded In-Circuit-Emulation.
- More system reliability features: watchdog, illegal write reset flag, mode protection for write error, watchdog protection for write error.

2.3 Applications

- Advanced educational toys
- High-end general STN/TFT LCD controller
- Kid storybook, E-book
- Hand-held, Multimedia LCD game
- Educational Learning Assistant
- Handheld organizer
- Data bank
- Multi Media Dictionary
- PDA

2.4 The Differences between GPL162002A and GPL162003A

- GPL162003A does not have TFT LCD feature. Other functions and pins are compatible with GPL162002A.

3 System Control

3.1 Introduction

This chapter describes the body information, reset option, system clock, system reliability, and operation mode. The features are depicted as follows:

- Body Information.
- Built-in 32768Hz/6MHz crystal circuit.
- Built-in 2 Phase-Lock Loop (PLL), one pumps from 32768Hz up to 12MHz, and the other pumps from 12MHz to 96MHz.
- Support clock driver in each mode, which can generate different kinds of speed in wild range.
- Level Low Voltage Reset (LVR).
- Build-in Watchdog Timer.
- Support wait mode, halt mode and sleep mode for power management.
- The clock of each device can be turned on/off individually to reduce the operating power.

System Control Register Summary Table

Name	Address	Description
P_BodyID	0x7800	Body Identification Number Register
P_CLK_Ctrl0	0x7804	Clock On/Off Control Register 0
P_CLK_Ctrl1	0x7805	Clock On/Off Control Register 1
P_Reset_Flag	0x7806	Reset Event Flag Register
P_Clock_Ctrl	0x7807	System Clock Control Register
P_LVR_Ctrl	0x7808	Low Voltage Reset Control Register
P_Watchdog_Ctrl	0x780A	Watchdog Control Register
P_Watchdog_Clear	0x780B	Watchdog Clear Register
P_WAIT	0x780C	Wait Mode Entrance Register
P_HALT	0x780D	Halt Mode Entrance Register
P_SLEEP	0x780E	Sleep Mode entrance Register
P_Power_State	0x780F	Current Power State Register
P_PLLN	0x7817	PLL's Divider selection
P_PLLWiatCLK	0x7818	PLL state change wait time
P_AD_Driving	0x781F	Address/Data Driving control Register

3.2 Device Identification

P_BodyID	0x7800							Body ID Number								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	0x8688															
Default	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BODYID	R	Body Identification Register For GPL162002 and GPL162003, the ID number is fixed to 0x8688.	0x8688 for GPL162002 and GPL162003

3.3 Reset Control

P_Reset_Flag		0x7806								Reset Event Flag Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	WDG	WDE	MPE	-	LVR
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
4	WDG	R/W	Watchdog Timeout Reset Flag	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
3	WDE	R/W	Watchdog Error Write Flag If programmers do NOT write 0xA005 to clear watchdog timer, this bit will be set to "1" by GPL162002A/162003A and CPU will be reset.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
2	MPE	R/W	Mode Protect Error Write Flag If programmers do NOT write 0x5005 to enter wait mode, or NOT write 0x500A to enter halt mode, or NOT write 0xA00A to enter standby mode, this bit will be set to "1" by GPL162002A/162003A, and CPU will be reset.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
1			Reserved	
0	LVR	R/W	Low Voltage Reset Flag If GPL162002 power is below designated threshold voltage, this flag will be set to "1" by GPL162002A/162003A, and CPU will be reset. The threshold voltage is defined in P_LVR_Ctrl.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

Programmers can confirm which type of reset is activated by reading the corresponding bit.

3.4 Clock Generation

There are two crystal circuits built in GPL162002A/162003A, which are for 32768Hz and 12MHz. When the built-in USB device/host function GPL162002A/162003A is used, it is recommended that a 12MHz crystal should be connected to GPL162002A/162003A to ensure that the error free 48MHz clock is generated. If the USB function is not used, users can choose to use 32768Hz crystal only. The selection between these two configurations is via the IC pin, BM2. If users pull high BM2 pin at start-up, the 12MHz and 32768Hz crystals will be used, otherwise, only the 32768 Hz crystal will be used. The following table shows the difference between these two configurations.

BM2	USB Function	Slow PLL (32768Hz -> 12MHz)	Fast PLL (12MHz -> 96MHz)
0	Not Active	Active	Active (After enter Fast mode)
1	Available when Fast PLL is on	Not Active	Active (After enter Fast mode)

The 32768Hz crystal must be connected to GPL162002A/162003A when USB device/host function is used, there will be two crystals connected to GPL162002A/162003A. The 12MHz crystal is used to generate Fast PLL from 12MHz to 96MHz, and 32768Hz crystal is used to trigger Real Time Clock unit and 32768Hz system clock when bit C32K in P_Clock_Ctrl is set to "1".

After power-on, the system will run at 12MHz system clock.

P_Clock_Ctrl						System Clock Control Register										
0x7807																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FAST	C32K	-	WEAK	-	C32KOFF	KCEN	-	-	-	-	DAPLLEN	CLK96M	CLKDIV		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FAST	R/W	Fast PLL Enable This control bit is used to enable internal Fast PLL logic circuit. 1. When C32K is set to 0 and this bit is set to 1, the Fast PLL will be enabled and generate 48 MHz clock. 2. When C32K is set to 0 and this bit is set to 0, the Fast PLL will be disabled and the system clock is coming from external 12MHz Crystal (when 12MHz crystal is used) or Slow PLL (from 32768Hz crystal to 12MHz PLL).	0= Disabled (12MHz) 1= Enabled (default 48MHz)

Bit	Function	Type	Description	Condition
14	C32K	R/W	CPU Clock Selection This control bit is used to select CPU clock between 32768Hz and PLL clock. When this bit is set to "1", the CPU clock will run at 32768Hz, and PLL will be turned off, no matter FAST bit is "1" or "0".	0= High speed clock, PLLCLK. 1= Low Speed clock, 32768Hz.
13			Reserved	
12	WEAK	R/W	32768 Hz Crystal Weak Mode Enable This bit is used to control the strong/ weak mode of 32768Hz crystal pad. After reset, the 32768Hz crystal pad will be set to strong mode to ensure that 32768Hz clock will start correctly. Users can choose to change the pad to weak mode to save power after power-on.	0= 32768 Hz Crystal Pad Operate in Strong Mode. 1= 32768 Hz Crystal Pad Operate in Weak Mode.
11			Reserved	
10	C32KOFF	R/W	IOB0 32768Hz Output Disable There will be a 32768 Hz output on IOB0; write 1 to this bit will turn off this output.	0= IOB0 output 32768 Hz. 1= IOB0 behaves as GPIO or other special function.
9	KCEN	R/W	IOB0 ~2 Key Change Interrupt Enable To turn on the key change wake-up function of these GPIO pins, programmers needs to write this bit to 1.	0= IOB0~2 key change function interrupt disable. 1= IOB0~2 key change function interrupt enable.
[8:5]			Reserved	
4	DAPLLEN	R/W	DA/AD PLL Enable Before turning on the DA or AD, programmers must set this bit to 1 and wait around 1ms until the PLL output is stable.	0: Disable DA/AD PLL. 1: Enable DA/AD PLL.
3	CLK96M	R/W	Current Clock Setting Register This bit is for USB function. The USB function needs 48MHz clock. When system clock is setting to 96MHz, this bit must be set to 1.	0: Current clock is not 96MHz. 1: Current clock is 96MHz.
[2:0]	CLKDIV		Clock Divide Selection. The clock divider operates under any kinds of configurations. It will divide the clock source selected by users and then output quotient as system clock. So the slowest clock in GPL162002 is 32768/128 = 256 Hz	000= SYSCLK = Clock Source 001= SYSCLK = Clock Source/2 010= SYSCLK = Clock Source/4 011= SYSCLK = Clock Source/8 100= SYSCLK = Clock Source/16 101= SYSCLK = Clock Source/32 110= SYSCLK = Clock Source/64 111= SYSCLK = Clock Source/128

The clock of each module can be turned on/off individually. This is done by writing the corresponding bits of the P_CLK_Ctrl0 and P_CLK_Ctrl1. If programmers write '1' to the corresponding bit of P_CLK_Ctrl0 and P_CLK_Ctrl1, the clock of the corresponding device will be turned on. If programmers write '0' to the corresponding bit of P_CLK_Ctrl0 and P_CLK_Ctrl1, the clock of the corresponding device will be turned off. Some important facts should be noted.

1. To turn off the clock of system bus and system control module is not allowed. This will cause unexpected results and cause the system crashed.
2. Before turning off the memory controller's clock, be sure you are not using the external bus for a program or placing data in the external bus.
3. Be sure the corresponding device is not active before turning off its clock.

P_CLK_Ctrl0		0x7804								Peripheral Clock Control Register0							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		Clock Source [15:0]															
Default		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

P_CLK_Ctrl1		0x7805										Peripheral Clock Control Register1					
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		Clock Source [31:16]															
Default		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Please refer to the follow table for the device of each source.

Clock Source	Device	Clock Source	Device
0	System Bus	16	3D surround
1	Memory	17	USB Host
2	GPIO	18	USB Device
3	Interrupt	19	IIC
4	Time Base	20	DMA
5	Timer A/B/C/D	21	SRC
6	DAC	22	EQ
7	Uart	23	SRAM 0
8	RTC	24	IIS DAC
9	SPI	25	Key Scan
10	Analog	26	MISC
11	LCD	27	--
12	--	28	TFT [*]
13	SP Bus	29	MP3
14	Timer E/F	30	System Control
15	SD/MMC	31	System Control

Note *: On GPL162003, this bit is invalidly.

P_PLLN	0x7817								Fast PLL output divider register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	PLLN						
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
[6:0]	PLLN	R/W	Fast PLL's output control This register can be changed only when system is not at FAST state. This means programmers need to switch PLL to slow mode then can change PLL clock. In other words, programmers must disable bit 15 of P_Clock_Ctrl (0x7807) first then change PLLN, enable bit 15 of P_Clock_Ctrl and polling P_State(0x780F) for PLL stable. The PLL system clock equals to PLLN multiplied by three.	0000000~000011: reserved 0000100: 12MHz 0000101: 15MHz 0000110: 18MHz 0000111: 21MHz 0001000: 24 MHz ... 0100000: 96MHz

3.5 System Reliability Control

P_LVR_Ctrl	0x7808								Low Voltage Reset Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	LVROFF	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:2]			Reserved	
1	LVROFF	R/W	Low Voltage Reset Off Selection This register is used to turn off the LVR reset when users do not wish to use the LVR function. The LVR reset flag is still set when LVR is 1. LVR reset voltage: 2.47~2.55V.	0= Enable 1= Disable
0			Reserved	

P_Watchdog_Ctrl	0x780A								Watchdog Reset Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WDGEN	WDGS	-	-	-	-	-	-	-	-	-	-	-	-	WDGPD	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	WDGEN	R/W	Write once to enable Watchdog reset	0= Disabled 1= Enabled
14	WDGS	W	Write once to select Reset Target For more information about system reset and CPU reset, refer to Appendix .	0= Reset System (included CPU) 1= Reset CPU
[13:3]			Reserved	
[2:0]	WDGPD	R/W	Watchdog period	000= 2 seconds 001= 1 second 010= 0.5 seconds 011= 0.25 seconds 1X0= 0.125 seconds 1X1= 62.5 seconds

P_Watchdog_Clear		0X780B		Watchdog Clear Register													
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WDGC															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	WDGC	W	Watchdog Clear Register Write A005 to clear watchdog timer only when watchdog is enabled (i.e. 0X780A [15] =1). Writing other value will reset CPU.	

3.6 Operation Mode Control

GPL162002A/162003A has three operation modes: Wait, Halt, and Sleep mode. Please refer to the following table. Note that these three modes will all yield CPU to be powered down.

	CPU	PLL (System Clock)	32768Hz Clock	After wakeup
Wait Mode	OFF	ON	ON	Next Instruction
Halt Mode	OFF	OFF	ON (RTC)	Reset CPU
Sleep Mode	OFF	OFF	OFF	System reset

When entering the halt mode, the system will disable PLL automatically so that it doesn't need to change system clock to 32768 Hz. And the system will run at the clock that set before entering halt mode when waking up from halt mode.

If the system enters halt mode, programmers can disable RTC to save more power. To determine CPU is whether power-on reset or wakeup from halt mode, refer to **chapter: Interrupt**.

P_WAIT	0x780C																Wait Mode Entrance Register																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
Function	WAIT																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																												
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0</

Bit	Function	Type	Description	Condition
[15:0]	WAIT	W	Wait Mode Entrance Register Write 0x5005 to enter wait mode (stop CPUCLK source only, but SYSCLK and 32768Hz are still valid).	

When writing 0x5005 to control register 0x780C to enter wait mode, programmers have to add at least 6 nop instructions to make sure the GPL162002A/162003A enter wait mode successfully.

P_HALT	0x780D															Halt Mode Entrance Register																													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																													
Function																HALT																													
Default																																													

Bit	Function	Type	Description	Condition
[15:0]	HALT	W	Halt Mode Entrance Register Write 0x500A to enter halt mode (stop CPUCLK and SYSCLK, but 32768Hz remains working). The RTC is still capable of running in this mode.	

P_Sleep	0x780E																Sleep Mode Entrance Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Function	SLEEP																															
Default																																

Bit	Function	Type	Description	Condition
[15:0]	SLEEP	W	Halt Mode Entrance Register Write 0xA00A to enter standby mode (stop all clock source: CPUCLK, SYSCLK, and 32768Hz). Once waking up from sleep mode, the system will be reset.	

P_State	0x780F								Power State Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-								State							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[15:4]	Reserved	R	Reserved	
[2:0]	State	R	Current Power State Register. Programmers can appreciate the current power state of GPL162002A/162003A by reading this register. Programmers cannot change system clock in changing state.	000= Clock Change State. 001= Slow State, Use Slow PLL clock or external 12MHz clock as clock source. 010= Fast State, Use Fast PLL as clock source. 011= 32768 State, Use 32768 Hz as clock source. 100= wait state. Use the previous setting before enter this state. 101= before stop state. Use 32768Hz as clock source, this state very short and halt or sleep state will entered. 110= halt state. Use 32768Hz as clock source, only a little logic is active. 111= sleep state. No clock source. Wake up by key-change.

Address/Data Bus Driving Strength Control

To meet the various requirements of external memory, in GPL162002A/162003A, it is capable of adjusting the driving ability of address and data bus. When external memory bus loading is small, reducing the driving ability will reduce the bouncing time of signals. When external memory loading is large, increasing the driving ability will reduce the signal delay.

P_AD_Driving

0x781F

Address/Data Driving Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	D_POFF	D_PH	D_SR	D_SMT	-	D_DRIVE		-	A_SR	A_SMT	-	A_DRIVE				
Default	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
15	D_POFF	R/W	Data Bus Automatic Pull High/Low Turn Off. This bit is used to control the automatic pull high/low function of data bus when entering HALT mode or STANDBY mode. This bit also controls the state of address bus when entering HALT mode or STANDBY mode.	0 = Automatic function is on. Address bus will be tied to high at the same time. 1 = Automatic function is off. Address bus will not be tied to high.
14	D_PH	R/W	Data Bus Automatic Pull High/Low Selection. This bit is used to select the automatic pull high/low function of data bus when entering HALT mode or STANDBY mode. This bit is	0 = Pull-low data bus when enter HALT/STANDBY mode. 1 = Pull-high bus when enter HALT/STANDBY mode.

Bit	Function	Type	Description	Condition
			valid only when D_POFF is 0.	
13	D_SR	R/W	Slew Rate control of data bus. This bit is used to control the slew rate of data bus.	0 = High slew rate of data bus. 1 = Low slew rate of data bus.
12	D_SMT	R/W	Schimit trigger of data bus. This bit is used to control the Schimit trigger of data bus.	0 = Turn-off the Schimit trigger of data bus. 1 = Turn-on the Schimit trigger of data bus.
[11]	Reserved	R	Reserved.	
[10:8]	D_DRIVE	R/W	Driving Strength of data Bus. This register is used to control the driving capability of data bus.	000 = 4 mA. 001 = 4 mA. 010 = 8 mA. 011 = 8 mA. 100 = 12 mA. 101 = 12 mA. 110 = 16 mA. 111 = 16 mA.
[7:6]	Reserved	R	Reserved	
5	A_SR	R/W	Slew Rate control of address bus. This bit is used to control the slew rate of address bus.	0 = High slew rate of address bus. 1 = Low slew rate of address bus.
4	A_SMT	R/W	Schimit trigger of address bus. This bit is used to control the Schimit trigger of address bus.	0 = Turn-off the Schimit trigger of address bus. 1 = Turn-on the Schimit trigger of address bus.
[3]	Reserved	R	Reserved.	
[2:0]	A_DRIVE	R/W	Driving Strength of Address Bus. This register is used to control the driving capability of address bus.	000 = 4 mA. 001 = 4 mA. 010 = 8 mA. 011 = 8 mA. 100 = 12 mA. 101 = 12 mA. 110 = 16 mA. 111 = 16 mA.

3.7 Special Note

GPL162002A/162003A supports three boot modes, for details, please refer to Memory Chapter. To select which boot mode is used, GPL162002A/162003A will detect pins, BM0 and BM1, at power-on or hardware reset cycle. When BM [1:0] = 00, the system will boot from external MCS0 memory. When BM [1:0] = 10, the system will boot from internal ROM. When BM [1:0] = 11 or 01, the system will boot from external EMUCE Memory instead of internal ROM.

For Generalplus Confidential Use Only
普格科技股份有限公司

4 Memory

4.1 Introduction

GPL162002A/162003A has a built-in internal ROM, SRAM and a NOR type flash memory controller with AMBA like interface. There are five chip select pins and one chip select pin for emulation, each memory device has 256 pages, and each page is 64K words. So, the controller can totally support up to 80M words for NOR type flash memories.

In addition to the ROM/SRAM/NOR type flash memory controller, the NAND type flash memory controller for 8 bits or 16 bits NAND type flash memories and SM (Smart Media) flash memories are available, and it supports H/W ECC (Error Correction Code).

4.2 FEATURE

- Support ROM / SRAM / NOR type flash memory.
- Five banks (5 chips select) are available for the supported memories.
- Each bank size is up to 256 pages, and each page is 64K words, the controller can totally support up to 80M words for NOR type flash memories.
- Supply the interface to access 8-bit or 16-bit NAND Flash memory.
- Support flexible Command/Address mode.
- Support Auto page Program/Read.
- Support single Program/Read by Firmware.
- Provide DMA interrupt request.
- Support NAND hardware ECC.
- Each memory can be configured as 8-bit mode access.

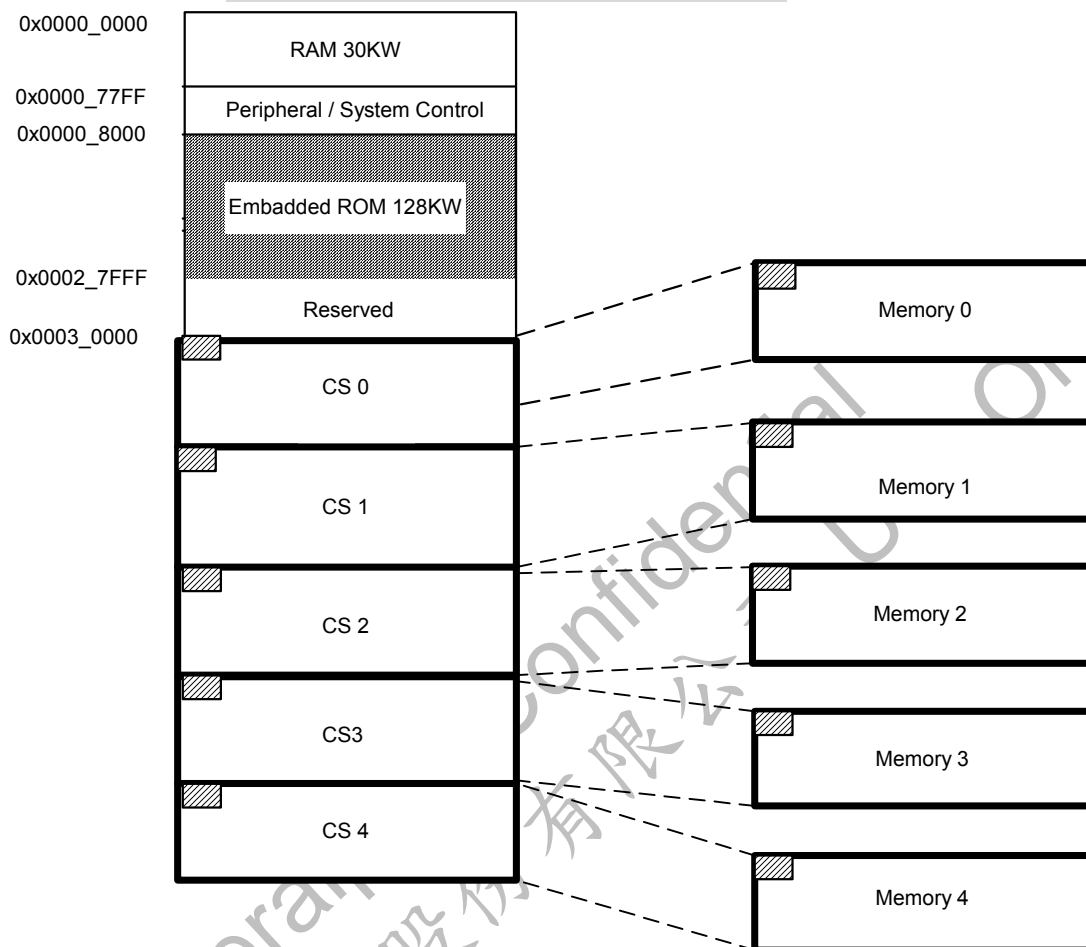
4.3 Memory Mappings

GPL162002A/162003A has a built-in 30K-word SRAM and a 128K-word internal ROM. Associated with external memory devices, GPL162002A/162003A is able to address up to 81920K-word locations. GPL162002A/162003A supports three boot modes, Boot from internal ROM, Boot from external ROM and boot EMU mode. The memory mappings of these three modes are as follows.

1. Internal ROM mode

When BM [1:0] = 10, GPL162002A/162003A will boot from internal embedded ROM. This mode is active at end product stage. The memory mapping of the internal ROM mode is shown below.

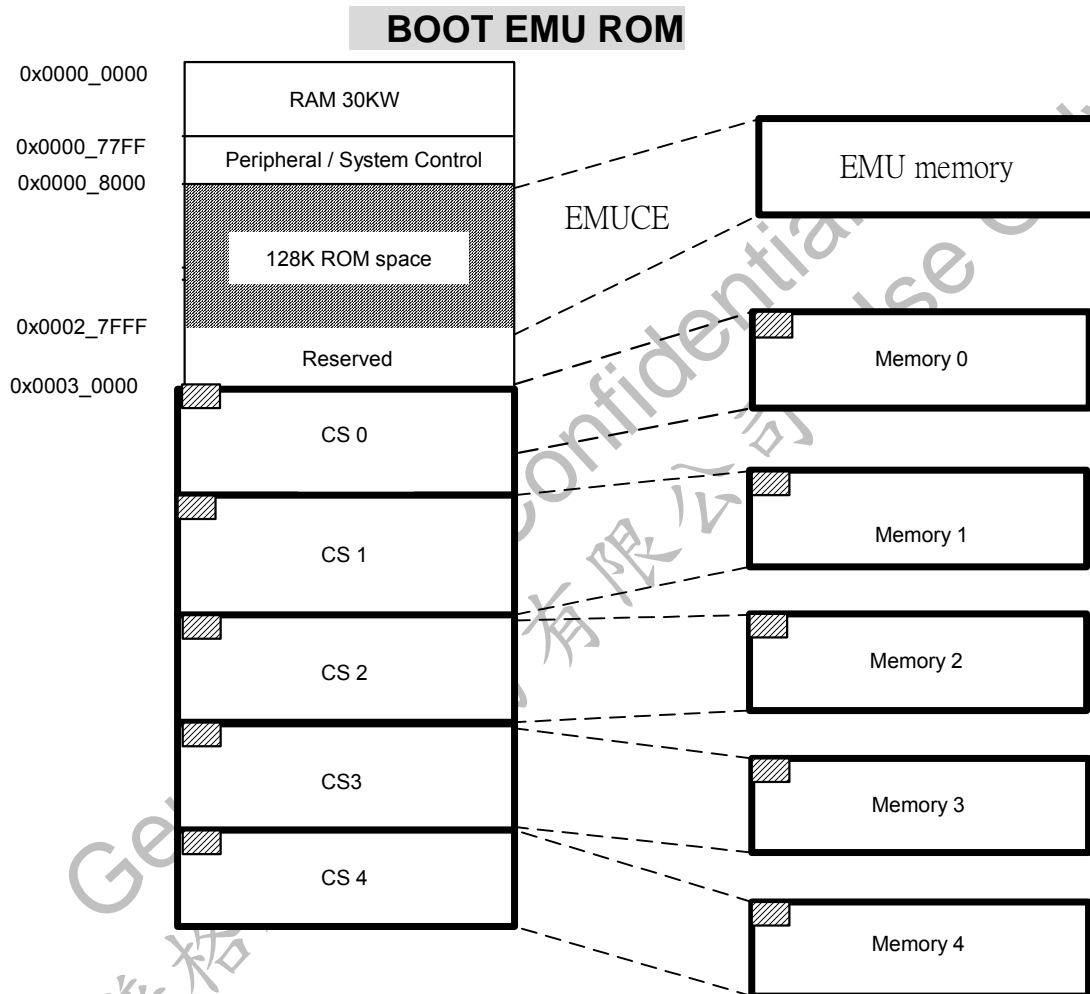
BOOT FROM INTERNAL ROM



2. EMU mode

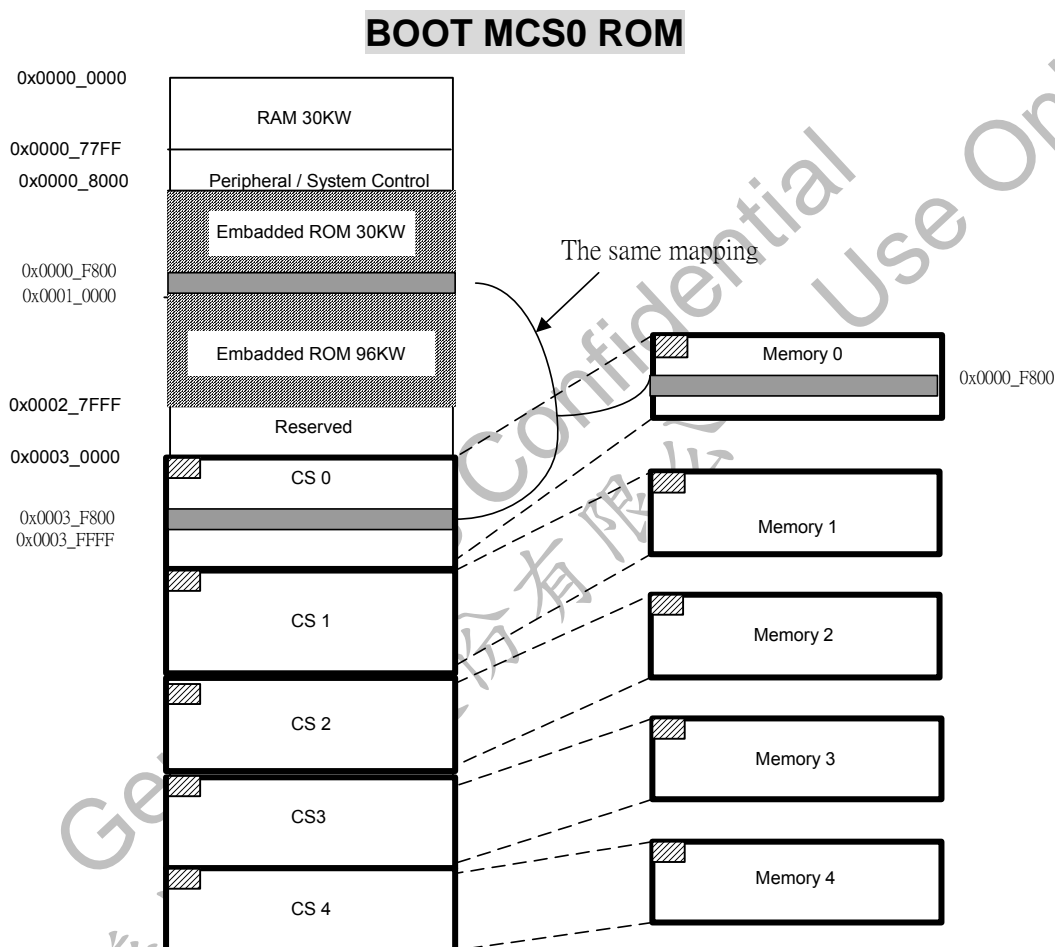
When BM [1:0] = 01 or 11, GPL162002A/162003A boot from external memory instead internal ROM.

In this mode, IC pin, BKCSB5 (EMUCE), will change to low when CPU access address from 0x8000 to 0x27FFF. The memory mapping of EMU mode is shown below.



3. MCS0 boot mode

When BM [1:0] = 00, GPL162002 will boot from external MCS0 memory. In this mode, default internal rom area 0xF800~0xFFFF are mapped to 0x3F800~0x3FFFF. We call these areas as CPU boot code area. And the area size can be adjusted dynamically by setting control register P_MAPSEL (0x7816). The memory mapping of MCS0 boot mode is shown below.



For GPL162002A/162003A, each page size is 64K-word. GPL162002A/162003A can access up to 256 pages (16384KW) of each memory controller. There are total five memory controllers on GPL162002A/162003A so that it can totally access up to 81920KW; in CPU view, however, it can only address up to 4MW. To access whole 81920KW, programmers need to use one control register to switch bank. Besides, GPL162002A/162003A supports five chip select signals to enable five external memory devices. Some of these chip select pins are shared with other special functions. For detail, please refer to **I/O Port**.

The 30K-word Internal SRAM (including stack) area is located in 0x000000 ~ 0x0077FF, and system

control registers and peripheral control registers reside in the 2K-word area, from 0x007800 to 0x007FFF.

Internal SRAM, system control register and peripheral control registers are positioned in the lower 32K-word of Page0. The layout of internal ROM starts from the upper 32K-word in page0, and ends up the upper 32K-word of Page2 (from 0x008000 to 0x027FFF). Therefore, internal resources will be stored in Page0, Page1, and Page2. Note that the higher 32K-word of Page2 is reserved (0x028000~0x02FFFF).

The address of the five external memory devices starts at 0x0003_0000, total 81920K-word. In other words, the size of five devices is programmable by software, but the addressing space of these five devices cannot be overlapped. Besides, if the system has to connect a NAND Flash, it will use one of these five chip select pins. It is recommended to use the system's last chip select pin.

See the following formula:

Size of (External Device0 + External Device1 + External Device2 + External Device3 + External Device4)
 $\leq 81920\text{K words}$

The method to define the address mapping of five external memory devices is to set up the size of each memory device. In GPL162002A/162003A, the start address of CS0 is 0x0003_0000. After the size of CS0 is given, the CS1 address is defined impliedly. That is 0x0003_0000 + size of CS0 (the unit is based on each 64K-word or saying 0x010000). After the CS1 is settled, the start address of CS2 is automatically defined, similar for CS3, CS4.

Example:

Suppose we have two 256K-Word SRAMs for CS0 and CS1 and A 512K-Word Flash memory and a 512K-Word ROM for CS2 and CS3, respectively.

Arrangement the following:

CS0: 0x0003_0000 ~ 0x006_FFFF as 256KW SRAM

CS1: 0x0007_0000 ~ 0x000A_FFFF as 256KW SRAM

CS2: 0x000B_0000 ~ 0x0013_FFFF as 512KW Flash

CS3: 0x0014_0000 ~ 0x001B_FFFF as 512KW ROM

In most cases, not all five external device will be used, nor all 24 address lines. Therefore, GPL162002A/162003A allows system designers to convert unused CS0 ~ CS4, Address signal (A17 ~ A23) or even read/write signal (#WE, #RE) to general purpose I/O. These signals are memory control signals in default after CPU resets. After configuring the control registers, these signals can be one of GPIOs. For details, please refer to Chapter: **I/O Ports**.

4.4 Memory Access Pin Configuration

Name	I/O	Description
MD[15:0]	I/O	Memory 16-bit data bus (Dedicated)
MA[16:0]	O	Memory address bus [16:0] (Dedicated)
MA[23:17]	O	Memory address bus [23:17] (shared with GPIO PortD[11:5])
CS[4:0]	O	Memory chip selection signal (shared with GPIO PortD[4:0])
WE	O	Memory write enable control signal (shared with GPIO PortB3)
RE	O	Memory read enable control signal (shared with GPIO PortB4)

4.5 Control Registers

Memory Control Register Summary Table

Name	Address	Description
P_MCS0_Ctrl	0x7820	Chip Selection 0 Memory Device Control Register
P_MCS1_Ctrl	0x7821	Chip Selection 1 Memory Device Control Register
P_MCS2_Ctrl	0x7822	Chip Selection 2 Memory Device Control Register
P_MCS3_Ctrl	0x7823	Chip Selection 3 Memory Device Control Register
P_MCS4_Ctrl	0x7824	Chip Selection 4 Memory Device Control Register
P_EMUCS_Ctrl	0x7825	EMU Chip Selection Memory Device Control Register
P_MCS_Byte_Sel	0x7826	CS0~CS4 and EMUCS Word/Byte Data Select
P_MCS3_WETimingCtrl	0x7827	MCS3 WE timing control register
P_MCS4_WETimingCtrl	0x7828	MCS4 WE timing control register
P_MCS3_RDTimingCtrl	0x7829	MCS3 RD timing control register
P_MCS4_RDTimingCtrl	0x782A	MCS4 RD timing control register
P_MCS3_TimingCtrl	0x782B	MCS3 CS timing control register
P_MCS4_TimingCtrl	0x782C	MCS4 CS timing control register
P_Mem_Ctrl	0x7840	Memory Control Register
P_Addr_Ctrl	0x7841	Memory A17~A25 Control Register
P_BankSwitch_Ctrl	0x7810	Bank Switch Control Register
P_MAPSEL	0x7816	CS0 boot mapping size select register

The P_MCS0_Ctrl, P_MCS1_Ctrl, P_MCS2_Ctrl, P_MCS3_Ctrl and P_MCS4_Ctrl are control registers for memory devices on chip select 0, 1, 2, 3, and 4. To make system operation more reliable, programmers should set appropriate wait cycles for each external device on these five control registers. The longer the wait period is, the more reliable a system is. The shorter the wait period is, the higher the performance is, but the less reliable a system is.

Wait state setup has one limitation. That is, device access time should **always** be smaller than designated wait cycle which is determined by number of system clocks. There is indeed access time information on memory device data sheet, but this access time criterion is under certain operating voltage and bus loading. Therefore, programmers should weigh some margins while trying to determine the period of wait cycle, especially when system voltage varies (not fixed to some specified operating voltages) and when there are too many memory devices in a system. (Bus loading)

P_MCS0_Ctrl	0x7820								CS0 Device Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS0SIZE								CS0MD		WARWAT		CS0WAIT			
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	CS0SIZE	R/W	Memory Device Size on Chip Select 0 (CS0), unit is 64K words. "CS0SIZE [7:0] + 1" defines the number of page for the entire memory device on CS0. Page size is 64K-word.	Range= 0 ~255 Size=(CS0SIZE[7:0] +1)* 64Kword Size Range= 64K-word ~ 16384K-word
[7:6]	CS0MD	R/W	CS0 Memory Device Access Mode. To define which memory device on CS0, such as ROM, SRAM, NOR or NAND Flash memories. If NAND Flash is selected, the MCS0 pin will keep low until it is changed to other memory type.	00= ROM/SRAM 01= ROM/SRAM 10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0]* SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	CS0WAIT	R/W	CS0 Memory Device Access Wait State Setup Criterion: ((CS0WAIT[3:0] + 1) * SYSCLK cycle) > memory device access time	Range= 0 ~15 Tw= (CS0WAIT[3:0] + 1) * SYSCLK

P_MCS1_Ctrl	0x7821								CS1 Device Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS1SIZE								CS1MD		WARWAT		CS1WAIT			
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	CS1SIZE	R/W	Memory Device Size on Chip Select 1 (CS1), unit is 64K words. "CS1SIZE [7:0] + 1" defines	Range= 0 ~255 Size=(CS1SIZE[7:0] +1)*

Bit	Function	Type	Description	Condition
			the number of page for the entire memory device on CS1. Page size is 64K-word.	64Kword Size Range= 64K-word ~ 16384K-word
[7:6]	CS1MD	R/W	CS1 Memory Device Access Mode To define which memory device on CS0, such as ROM, SRAM, NOR or NAND Flash memories. If NAND Flash is selected, the MCS1 pin will keep low until it is changed to other memory type.	00= ROM/SRAM 01= ROM/SRAM 10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0*] SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	CS1WAIT	R/W	CS1 Memory Device Access Wait State Setup Criterion: $((CS1WAIT[3:0] + 1) * SYSCLK \text{ cycle})$ > memory device access time	Range= 0 ~15 $Tw = (CS1WAIT[3:0] + 1) * SYSCLK$

P MCS2 Ctrl
0x7822
CS2 Device Control Register

CS2		CS2SIZE								CS2 Waiter Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS2SIZE								CS2MD		WARWAT		CS2WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	CS2SIZE	R/W	Memory Device Size on Chip Select 2 (CS2), unit is 64K words. "CS2SIZE [7:0] + 1" defines the number of page for the entire memory device on CS2. Page size is 64K-word.	Range= 0 ~255 Size=(CS2SIZE[7:0] +1)* 64Kword Size Range= 64K-word ~ 16384K-word
[7:6]	CS2MD	R/W	CS2 Memory Device Access Mode To define which memory device on CS2, such as ROM, SRAM, NOR or NAND Flash memories. If NAND Flash is selected, the MCS2 pin will keep low until it is changed to other memory type.	00= ROM/SRAM 01= ROM/SRAM 10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0*] SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	CS2WAIT	R/W	CS2 Memory Device Access Wait State Setup Criterion: $((CS2WAIT[3:0] + 1) * SYSCLK \text{ cycle})$	Range= 0 ~15 $Tw = (CS2WAIT[3:0] + 1) * SYSCLK$

Bit	Function	Type	Description	Condition
			> memory device access time	

P_MCS3_Ctrl
0x7823
CS3 Device Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS3SIZE								CS3MD		WARWAT		CS3WAIT			
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	CS3SIZE	R/W	Memory Device Size on Chip Select 3 (CS3), unit is 64K words. "CS3SIZE [7:0] + 1" defines the number of page for the entire memory device on CS3. Page size is 64K-word.	Range= 0 ~ 255 Size=(CS3SIZE[7:0] + 1)* 64Kword Size Range= 64K-word ~ 16384K-word
[7:6]	CS3MD	R/W	CS3 Memory Device Access Mode To define which memory device on CS3, such as ROM, SRAM, NOR or NAND Flash memories. If NAND Flash is selected, the MCS3 pin will keep low until it is changed to other memory type.	00= ROM/SRAM 01= ROM/SRAM 10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0]* SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	CS3WAIT	R/W	CS3 Memory Device Access Wait State Setup Criterion: (CS3WAIT[3:0] + 1) * SYSCLK cycle > memory device access time	Range= 0 ~15 Tw= (CS3WAIT[3:0] + 1) * SYSCLK

P_MCS4_Ctrl
0x7824
CS4 Device Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CS4SIZE								CS4MD		WARWAT		CS4WAIT			
Default	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]	CS4SIZE	R/W	Memory Device Size on Chip Select 4(CS4), unit is 64K words. "CS4SIZE [7:0] + 1" defines the number of page for the entire memory device on CS4. Page size is 64K-word.	Range= 0 ~255 Size=(CS4SIZE[7:0]+1)*64K word SizeRange=64K-word~ 16384K-word
[7:6]	CS4MD	R/W	CS4 Memory Device Access Mode To define which memory device on CS4,	00= ROM/SRAM 01= ROM/SRAM

Bit	Function	Type	Description	Condition
			such as ROM, SRAM, NOR or NAND Flash memories. If NAND Flash is selected, the MCS4 pin will keep low until it is changed to other memory type.	10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0] * SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	CS4WAIT	R/W	CS4 Memory Device Access Wait State Setup Criterion: $((CS4WAIT[3:0] + 1) * SYSCLK \text{ cycle})$ > memory device access time	Range= 0 ~15 $Tw = (CS4WAIT[3:0] + 1) * SYSCLK$

P_EMUCS_Ctrl
0x7825
EMU Device Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-								EMCMD		WARWAT		EMUCSWAIT			
Default	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]			Reserved.	
[7:6]	EMCMD	R/W	EMU Memory Device Access Mode To define which memory device is on EMUCE, such as ROM, SRAM, NOR or NAND Flash memories.	00= ROM/SRAM 01= ROM/SRAM 10= NOR Flash 11= NAND Flash
[5:4]	WARWAT	R/W	SRAM Write after Read Wait State When data is written to memory, and then read it immediately from the same address, CPU will wait WARWAT [1:0] * SYSCLK to read it.	WARWAT[1:0]*SYSCLK
[3:0]	EMUCS WAIT	R/W	EMU Memory Device Access Wait State Setup Criterion: $((EMUCSWAIT[3:0] + 1) * SYSCLK \text{ cycle})$ > memory device access time	Range= 0 ~15 $Tw = (EMUCSWAIT[3:0] + 1) * SYSCLK$

GPL162002A/162003A allows programmers to define wait cycles for each external device by setting corresponding CS0WAIT, CS1WAIT, CS2WAIT, CS3WAIT, CS4WAIT and EMUCSWAIT control bits. The default wait cycles are 16 system clock cycles to ensure system working reliably without any setup.

GPL162002A/162003A can support access 8-bit memory mode via setup P_MCS_Byte_Sel register. The 8-bit mode for each chip select is based on bit [4:0] of P_MCS_Byte_Sel. In this mode, only low

byte data is effective and high byte data will be filled all zeros.

P_MCS_Byte_Sel	0x7826										MCS Word/Byte Data Select register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-										EMU	S4	S3	S2	S1	S0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5]	EMU	R/W	If This bit is set 1, then EMUCS accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit
[4]	S4	R/W	If This bit is set 1, then CS4 accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit
[3]	S3	R/W	If This bit is set 1, then CS3 accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit
[2]	S2	R/W	If This bit is set 1, then CS2 accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit
[1]	S1	R/W	If This bit is set 1, then CS1 accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit
[0]	S0	R/W	If This bit is set 1, then CS0 accesses data in 8-bit mode. Only low byte available. High byte will be filled zero.	0: 16-bit 1: 8-bit

P_MCS3_WETimingCtrl	0x7827										MCS3 WE timing control register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													WEB3NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	WEB3NUM	R/W	CSB3 and WEB3 Program Timing Register.	Range= 0 ~15 Tw= WEB3NUM[3:0] * SYSCLK

P_MCS4_WETimingCtrl	0x7828										MCS4 WE timing control register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													WEB4NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	WEB4NUM	R/W	CSB4 and WEB4 Program Timing Register.	Range= 0 ~15 Tw= WEB4NUM[3:0] * SYSCLK

P_MCS3_RDTimingCtrl 0x7829
MCS3 RD timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													RDB3NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	RDB3NUM	R/W	CSB3 and RDB3 Program Timing Register.	Range= 0 ~15 Tw= RDB3NUM[3:0] * SYSCLK

P_MCS4_RDTimingCtrl 0x782A
MCS4 RD timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													RDB4NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	RDB4NUM	R/W	CSB4 and RDB4 Program Timing Register.	Range= 0 ~15 Tw= RDB4NUM[3:0] * SYSCLK

P_MCS3_TimingCtrl 0x782B
MCS3 CS timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													CSB3NUM			
Init	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	CSB3NUM	R/W	CSB3 Program Timing Register.	Range= 0 ~15 Tw= CSB3NUM[3:0] * SYSCLK

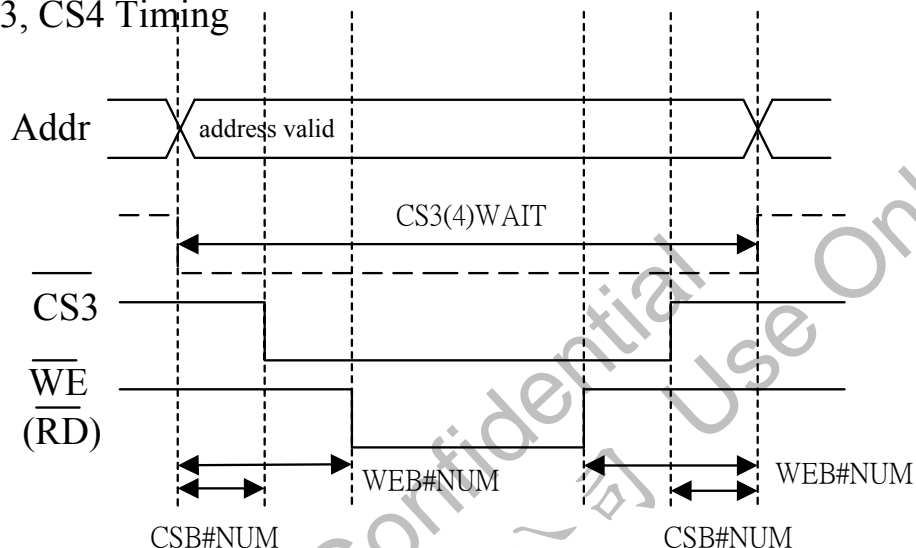
P_MCS4_TimingCtrl 0x782C
MCS4 CS timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function													CSB4NUM			
Init	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

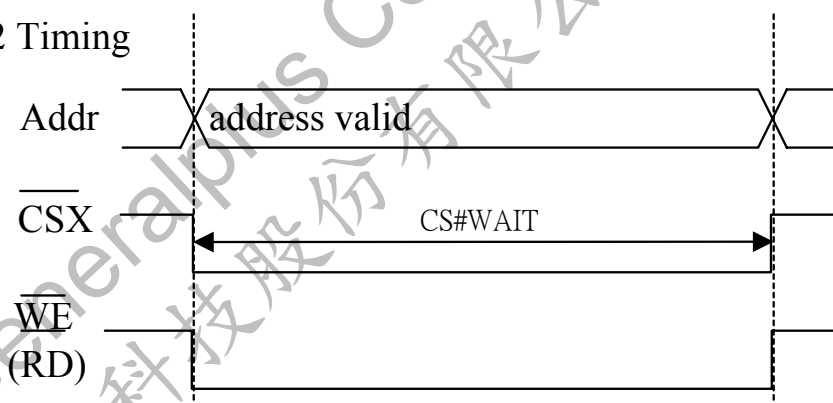
Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	CSB4NUM	R/W	CSB4 Program Timing Register.	Range= 0 ~15 Tw= CSB4NUM[3:0] * SYSCLK

Among all the chip select signals, CS3 and CS4 access setup time and hold time can be adjusted. This makes CS3 and CS4 more flexible for special memory devices, such as Compact Flash cards. Please refer to the following timing diagrams.

CS3, CS4 Timing



CS0~CS2 Timing



Following control registers are used to change the memory address signals and memory control signals to GPIO functions. After power-on reset, the default settings of these pins are as memory access signals, except CS4. If there is no external memory device, or external devices are less than five, some of memory addresses or memory control signals can be used as GPIOs. To change these pins to GPIOs, programmers must write certain word to a corresponding control register.

P_Mem_Ctrl			0x7840								Memory Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	WE	RD	-	MCS4	MCS3	MCS2	MCS1	MCS0
Default	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:14]			Reserved	
[13:8]			These bits have to be set as "0."	
7	WE	R/W	Memory Write Enable Signal WEB. When write 0 to this bit, PortB [3] becomes GPIO. Otherwise, PortB [3] remains Write Enable Signal.	0= Disable 1= Enable
6	RD	R/W	Memory Read Enable Signal OEB. When write 0 to this bit, PortB [4] becomes GPIO. Otherwise, PortB [4] remains Read Enable Signal.	0= Disable 1= Enable
5			Reserved	
4	MCS4	R/W	Memory chip select 4 enable pin. MCS4 is shared with IOD[4].	0= Disable 1= Enable
3	MCS3	R/W	Memory chip select 3 enable pin. MCS3 is shared with IOD[3].	0= Disable 1= Enable
2	MCS2	R/W	Memory chip select 2 enable pin. MCS2 is shared with IOD[2].	0= Disable 1= Enable
1	MCS1	R/W	Memory chip select 1 enable pin. MCS1 is shared with IOD[1].	0= Disable 1= Enable
0	MCS0	R/W	Memory chip select 0 enable pin. MCS0 is shared with IOD[0].	0= Disable 1= Enable

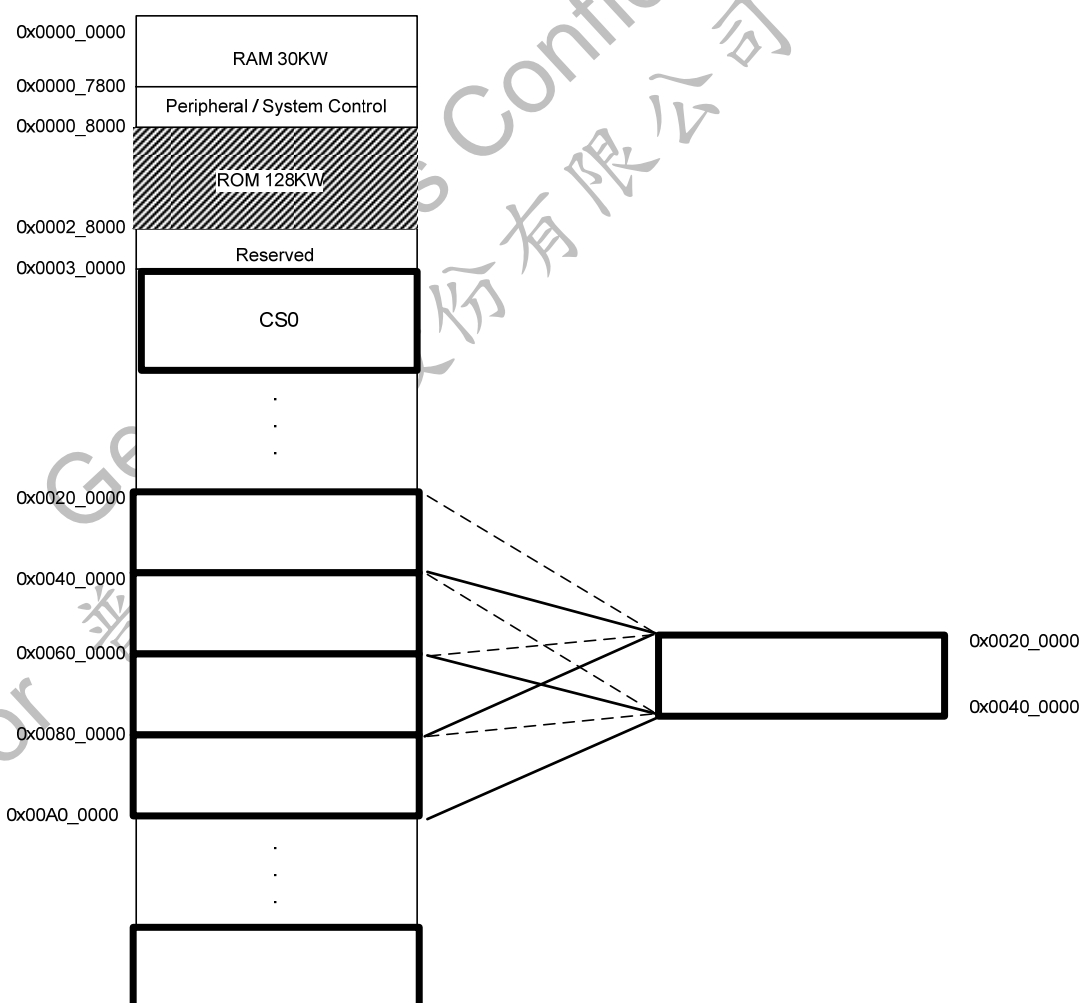
P_Addr_Ctrl			0x7841								Memory Address A17~A25 Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	MA23	MA22	MA21	MA20	MA19	MA18	MA17
Default	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	MA23	R/W	Address bus MA23 enable pin. MA23 is shared with IOD[11].	0= Disable 1= Enable
5	MA22	R/W	Address bus MA22 enable pin. MA22 is shared with IOD[10].	0= Disable 1= Enable
4	MA21	R/W	Address bus MA21 enable pin. MA21 is shared with IOD[9].	0= Disable 1= Enable
3	MA20	R/W	Address bus MA20 enable pin. MA20 is shared with IOD[8].	0= Disable 1= Enable

Bit	Function	Type	Description	Condition
2	MA19	R/W	Address bus MA19 enable pin MA19 is shared with IOD[7]	0= Disable 1= Enable
1	MA18	R/W	Address bus MA18 enable pin MA18 is shared with IOD[6]	0= Disable 1= Enable
0	MA17	R/W	Address bus MA17 enable pin MA17 is shared with IOD[5]	0= Disable 1= Enable

4.6 Bank Switch Control

As mentioned in the previous sections, GPL162002A/162003A is able to access totally 81920KW. However, in CPU view, it can only address up to 4MW, 0x000000 ~ 0x3FFFFFF. To use the address space larger than 0x3FFFFFF, programmers need to set bank switch control register before accessing the space. The address from 0x0020_0000 to 0x003F_FFFF is the memory space used to switch bank, and each bank size is 2MW.



P_BankSwitch_Ctrl 0x7810
Bank Switch Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-										Bank					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[15:6]	Reserved	R	Reserved	
[5:0]	Bank	W	Bank Number Register When these bits are set to 0x01, Physical address 0x200000~0x3FFFFF will be mapped to 0x20000~0x3FFFFF. When these bits are set to 0x02, Physical address 0x400000~0x5FFFFF will be mapped to 0x20000~0x3FFFFF. And so on.	

When BM [1:0] is set to 2b'00, the external MCS0 boot mode is selected. The mapping size of external boot area can be changed by setting P_MAPSEL register.

P_MAPSEL
0x7816
CS0 boot mapping size select register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-													MAPSEL		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]	Reserved	R	Reserved	
[2:0]	MAPSEL	W	CS0 boot map size register. When programmers use MCS0 boot mode, Generalplus suggests programmers set this register immediately after CPU starts to run.	000: 2K (0xF800~0xFFFF) is mapping to 0x3F800~0x3FFFF 001: 4K (0xF000~0xFFFF) is mapping to 0x3F000~0x3FFFF 010: 6K (0xE800~0xFFFF) is mapping to 0x3E800~0x3FFFF 011: 8K (0xE000~0xFFFF) is mapping to 0x3E000~0x3FFFF 100: 10K (0xD800~0xFFFF) is mapping to 0x3D800~0x3FFFF 101: 12K (0xD000~0xFFFF) is mapping to 0x3D000~0x3FFFF 110: 16K (0xC000~0xFFFF) is mapping to 0x3C000~0x3FFFF 111: 32K (0x8000~0xFFFF) is mapping to 0x38000~0x3FFFF

4.7 Vectors

Interrupt Vector	
Address	Function
0x00FFF5	Software Break
0x00FFF6	Fast IRQ
0x00FFF7	RESET
0x00FFF8	IRQ0
0x00FFF9	IRQ1
0x00FFFA	IRQ2
0x00FFFB	IRQ3
0x00FFFC	IRQ4
0x00FFFD	IRQ5
0x00FFFE	IRQ6
0x00FFFF	IRQ7

4.8 Stack Location

Generalplus recommends that stack starts at the end of Internal SRAM; that is, stack pointer is set to 0x0077FF. Stack area will grow from bottom to top.

4.9 Chip Select (Project Setting) on IDE

SUNPLUS u'nSP IDE provides a setting dialog box to set up external CS [4:0] configuration. In this way, programmers can set up the size, mode and wait state of CS [4:0] easily and quickly. Moreover, programmers can review start and end addresses of CS [4:0] memory on other dialog boxes after settings.

Note that the setting on dialogue box is only for downloading procedure, not for the normal operation of GPL162002A/162003A while it is running. Therefore, programmers should give appropriate software codes to set P_MCS0_Ctrl, P_MCS1_Ctrl, P_MCS2_Ctrl, P_MCS3_Ctrl, and P_MCS4_Ctrl registers in a project.

Generalplus recommends programmers should set this dialog box before downloading programs; otherwise, ICE download function is possibly failed. Furthermore, the dialogue box mentioned above can also determine which memory device (CS0, CS1, CS2, CS3, and CS4) will be downloaded in order to save downloading time.

For Detail user manual on IDE, refer to **appendix: Project Setting on IDE**.

4.10 Program Examples

R1 = 0x0041 // SRAM, 64KW, 35ns access time
[P_MCS0_Ctrl] = R1 // External CS0 memory wait state = +1 system cycle
// CPU memory mapping is 0x003_0000 ~ 0x003_FFFF

R1 = 0x0142 // SRAM, 128KW, 50ns access time
[P_MCS1_Ctrl] = R1 // External CS1 memory wait state = +2 system cycle
// CPU memory mapping is 0x0004_0000 ~ 0x0005_FFFF

R1 = 0x0383 // FLASH, 256KW, 70ns access time
[P_MCS2_Ctrl] = R1 // External CS2 memory wait state = +3 system cycle
// CPU memory mapping is 0x006_0000 ~ 0x009_FFFF

R1 = 0x00CF // NAND Flash
[P_MCS3_Ctrl] = R1

5 I/O Ports

5.1 Available Ports

The purpose of General-Purpose-Input-Output (GPIO) is to communicate with other devices. Four programmable I/O ports are available in GPL162002A/162003A: PortA, PortB, PortC, and PortD. Each I/O pin on these 4 ports can be bit-by-bit configured by software. Almost every I/O pin on these 4 ports can be programmed as special function. In other words, many special function control signals share with I/O ports. Besides, the PortB [2:0] provide wake-up capability.

To change these 4 ports from GPIO functions to special functions, programmers need to enable the corresponding special functions. This is because special functions have higher priority than GPIO does. When special functions are activated, GPIO function will be disabled and any setting on GPIO will become invalid. To change each memory control signal to GPIO function, programmers have to set its corresponding write-once register for each I/O pin. Write-once mechanism on these registers is for system reliability purpose.

Following table depicts shared information about I/O ports and their special functions.

Note: The PortA TFT D[15..0] and PortC[3..0] TFT control signals are invalid on GPL162003 body.

PortA and Special Functions Shared Information

	15	14	13	12	11	10	9	8
Special Function	Key Scan							
Signal	KEYIN7	KEYIN6	KEYIN5	KEYIN4	KEYIN3	KEYIN2	KEYIN1	KEYIN0
In/Out	I	I	I	I	I	I	I	I
Special Function	LCD(TFT) interface D[15:8]							
Signal	TFT_D15	TFT_D14	TFT_D13	TFT_D12	TFT_D11	TFT_D10	TFT_D9	TFT_D8
In/Out	O	O	O	O	O	O	O	O

	7	6	5	4	3	2	1	0
Special Function	Key Scan							
Signal	KEYOUT7	KEYOUT6	KEYOUT5	KEYOUT4	KEYOUT3	KEYOUT2	KEYOUT1	KEYOUT0
In/Out	O	O	O	O	O	O	O	O
Special Function	LCD(TFT/STN) Interface D[7:0]							
Signal	LCD_D7	LCD_D6	LCD_D5	LCD_D4	LCD_D3	LCD_D2	LCD_D1	LCD_D0
In/Out	O	O	O	O	O	O	O	O

PortB and Special Functions Shared Information

	15	14	13	12	11	10	9	8
Special Function	Touch Panel Interface				Analog Input		Nand Flash Interface	
Signal	TSMY	TSMX	TSPY	TSPX	Line4	Line3	NF_RDY	NF_ALE
In/Out	I	O	O	O	O	O	O	O
Special Function					SPI Interface			
Signal					SPI_CLK	SPI_CS		
In/Out					I/O	I		

	7	6	5	4	3	2	1	0
Special Function	Nand Flash Interface			Memory Signals		Timer Specified Output		
Signal	NF_CLE	NF_OEB	NF_WEB	OEB	WEB	CCPC	CCPB	CCPA
In/Out	O	O	O	O	O	I/O	I/O	I/O
Special Function						Key Change		
Signal						Key_Ch2	Key_Ch1	Key_Ch0
In/Out						I	I	I

PortC and Special Functions Shared Information

	15	14	13	12	11	10	9	8
Special Function	DAC IIS Interface				SDC IF	UART/IrDA Interface		SDC IF
Signal	IIS_MCLK	IIS_BCLK	IIS_LRCK	IIS_SD	SD_DAT3	UART_RX	UART_TX	SD_DAT2
In/Out	O	O	O	O	I/O	I	O	I/O
Special Function	IIC Interface							
Signal			IIC_DAT	IIC_CLK				
In/Out			I/O	O				

	7	6	5	4	3	2	1	0
Special Function	SDC Interface				LCD Interface			
Signal	SD_DAT1	SD_DAT0	SD_CMD	SD_CLK	STN_CP TFT_Vsync	STN_LP TFT_Hsync	STN_FP TFT_DE	STN_FM TFT_CLK
In/Out	I/O	I/O	O	O	O	O	O	O

PortD and Special Functions Shared Information

	15	14	13	12	11	10	9	8
Special Function	General purpose		Extnal Input		Memory Address Bus			
Signal	IOD15	IOD14	EXTB	EXTA	MA23	MS22	MS21	MS20
In/Out	I/O	I/O	I	I	O	O	O	O
Special Function					SPI IF			
Signal					SPI_DI			
In/Out					I			

	7	6	5	4	3	2	1	0
Special Function	Memory Address Bus			Memory Chip Selection				
Signal	MA19	MA18	MA17	MCS4	MCS3	MCS2	MCS1	MCS0
In/Out	O	O	O	O	O	O	O	O
Special Function				SPI IF				
Signal				SPI_DO				
In/Out				O				

IOD12 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
External Interrupt A	Floating With pull-low resistor With pull-high resistor	P_MINT_Ctrl.bit0=1	When one of these modes is enabled, direction control bit (P_IOD_Dir.bit12) is forced to "0". (That is, original content of direction control bit will be ignored). Programmers can configure 3 input modes by modifying corresponding data and attribution control bit directly (P_IOD_Data.bit12, P_IOD_Attrib.bit12). 00 = with pull-low resistor 01 = with pull-high resistor 1X = floating
TimerX Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerX_Ctrl.bit[3:0]=7	
GPIO	All modes		

IOD13 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
External Interrupt B	Floating With pull-low resistor With pull-high resistor	P_MINT_Ctrl.bit1=1	When one of these modes is enabled, direction control bit (P_IOD_Dir.bit13) is forced to "0". (That is, original content of direction control bit will be ignored). Programmers can configure 3 input modes by modifying corresponding data and attribution control bit directly (P_IOD_Data.bit13, P_IOA_Attrib.bit13). 00 = with pull-low resistor 01 = with pull-high resistor 1X = floating
TimerX Clock Source	Floating With pull-low resistor With pull-high resistor	P_TimerX_Ctrl.bit[6:4]=7	
GPIO	All modes		

IOB0 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
TimerA Capture Mode Trigger Input	Floating	P_TimerA_CCP_Ctrl. bit[15:14]=01	When one of these modes is enabled, direction / attribution / data control bits are forced to corresponding mode. The contents on direction / attribution / data control bits have no effect on this I/O pad.
TimerA Comparison Mode Event Output	Output Buffer (High or Low)	P_TimerA_CCP_Ctrl. bit[15:14]=10	
TimerA PWM Mode Signal Output	Output Buffer (High or Low)	P_TimerA_CCP_Ctrl. bit[15:14]=11	
Key change0	Input pull high or low	P_MINT_Ctrl bit.10=1	
GPIO	All modes		

IOB1 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
TimerB Capture Mode Trigger Input	Floating	P_TimerB_CCP_Ctrl. bit[15:14]=01	When one of these modes is enabled, direction / attribution / data control bits are forced to corresponding mode. The contents on direction / attribution / data control bits have no effect on this I/O pad.
TimerB Comparison Mode Event Output	Output Buffer (High or Low)	P_TimerB_CCP_Ctrl. bit[15:14]=10	
TimerB PWM Mode Signal Output	Output Buffer (High or Low)	P_TimerB_CCP_Ctrl. bit[15:14]=11	
Key change1	Input pull high or low	P_MINT_Ctrl bit.12=1	
GPIO	All modes		

IOB2 Pin Special Function Shared Information

Special Function	I/O Mode Supported	Enable Control bit	Description
TimerC Capture Mode Trigger Input	Floating	P_TimerC_CCP_Ctrl. bit[15:14]=01	When one of these modes is enabled, direction / attribution / data control bits are forced to corresponding mode. The contents on direction / attribution / data control bits have no effect on this I/O pad.
TimerC Comparison Mode Event Output	Output Buffer (High or Low)	P_TimerC_CCP_Ctrl. bit[15:14]=10	
TimerC PWM Mode Signal Output	Output Buffer (High or Low)	P_TimerC_CCP_Ctrl. bit[15:14]=11	
Key change2	Input pull high or low	P_MINT_Ctrl bit.14=1	
GPIO	All modes		

5.2 General Purpose I/Os Configuration

GPL162002A/162003A provides a bit-to-bit I/O configuration; every I/O configuration can be defined individually. To set up a bit configuration, three control registers must be setup: Data, Attribution, and Direction. The following table is a summary of I/O configuration setting. Each corresponding bit in these three control registers should be given a value to set one bit configuration. For example, suppose PortA.0 is used as an input port with internal pull-low resistors. The bit0 in PortA DIRECTION, ATTRIBUTION and DATA control registers should be given "000" in binary. If PortA.1 is used as a floating input port with wake-up functions, the bit1 in PortA DIRECTION, ATTRIBUTION and DATA control registers should be given "010" in binary.

Reading operation on I/O DATA control registers will read status from external I/O pads. On the other hand, GPL162002A/162003A also provides a control register, I/O BUFFER. This control register holds the setting data (value) that is previously written into I/O DATA control register. Therefore, programmers do not need an extra variable (1-word SRAM) to store (hold) the previous setting on DATA control register.

5.3 General Purpose I/Os Function Table

Direction Register	Attribution Register	Data Register	Function	Wakeup	Description
0	0	0	Pull Low*	Yes**	Input with pull low
0	0	1	Pull High	Yes**	Input with pull high
0	1	0	Float	Yes**	Float (High Impedence)
0	1	1	Float	No	Float (High Impedence)
1	0	0	Output High	No	Output with buffer (inverted -content of buffer register)
1	0	1	Output Low	No	Output with buffer (inverted -content of buffer register)
1	1	0	Output Low	No	Output with buffer
1	1	1	Output High	No	Output with buffer

* Default is input mode with pull-low state.

** Only PortA and PortB in the state of "000", "001" and "010" have wake-up capability.

ODP output configuration can be done in the way of changing between float state ("011" in binary) and output high state ("111" in binary) by only modifying the direction bit from "0" to "1". Similarly, **ODN output configuration** can be done in the way of changing between float state ("010" in binary) and output low state ("110" in binary) by only modifying the direction bit from "0" to "1".

5.4 Control Register

I/O Port Control Register Summary Table

Name	Address	Description
P_IOA_Data	0x7860	I/O PortA Data Register
P_IOA_Buffer	0x7861	I/O PortA Buffer Register
P_IOA_Dir	0x7862	I/O PortA Direction Register
P_IOA_Attrib	0x7863	I/O PortA Attribution Register
P_IOB_Data	0x7868	I/O PortB Data Register
P_IOB_Buffer	0x7869	I/O PortB Buffer Register
P_IOB_Dir	0x786A	I/O PortB Direction Register
P_IOB_Attrib	0x786B	I/O PortB Attribution Register
P_IOB_Latch	0x786C	I/O PortB Latch Register for Wakeup
P_IOC_Data	0x7870	I/O PortC Data Register
P_IOC_Buffer	0x7871	I/O PortC Buffer Register
P_IOC_Dir	0x7872	I/O PortC Direction Register
P_IOC_Attrib	0x7873	I/O PortC Attribution Register
P_IOD_Data	0x7878	I/O PortD Data Register
P_IOD_Buffer	0x7879	I/O PortD Buffer Register
P_IOD_Dir	0x787A	I/O PortD Direction Register
P_IOD_Attrib	0x787B	I/O PortD Attribution Register

P_IOA_Data		0x7860						IOA Data Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOADATA	R/W	Executing the writing operation in this register will latch setup value into I/O PortA data register. Similarly, executing the read operation in this register will read the status from I/O PortA external pads.	Refer to the above table, I/O port configuration and function.

P_IOA_Buffer		0x7861								IOA Buffer Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOABUF															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOABUF	R/W	Executing the read operation in this register will read the setup value from I/O PortA data register, which is previously latched by IOADATA writing operation.	IOABUF(R) =IOADATA(W) =IOABUF(W)

P_IOA_Dir
0x7862
IOA Direction Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOADIR	R/W	This control register sets the direction of I/O PortA. In addition, the direction setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOA_Attrib
0x7863
IOA Attribution Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOAATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOAATT	R/W	This control register defines the attribution of I/O PortA. In addition, the attribution setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOB_Data
0x7868
IOB Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOBDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOBDATA	R/W	Executing the writing operation in this register will latch setup value into I/O PortB data register. Similarly, executing the read operation in this register will read the status from I/O PortB external pad.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOB_Buffer		0x7869								IOB Buffer Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBBUF															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOBBUF	R/W	Executing the read operation in this register will read the setup value from I/O PortB data register, which is previously latched by IOBDATA writing operation.	IOBBUF(R) =IOBDATA(W) =IOBBUF(W)

P_IOB_Dir		0x786A								IOB Direction Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBDIR															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOBDIR	R/W	This control register sets the direction of I/O PortB. In addition, the direction setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOB_Attrib		0x786B								IOB Attribution Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBATT															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOBATT	R/W	This control register defines the attribution of I/O PortB. In addition, the attribution setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOB_Latch		0x786C								IOB Latch for Key Change Wakeup							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	IOBLHW		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
[2:0]	IOBLHW	R	This control register latches the I/O PortB status for key-changed wake-up purpose. Wake-up is triggered if any I/O state of PortB is different from at the time latched. This latch operation must be done before entering sleep mode (for more information, refer to Chapter: System Control).	

P_IOC_Data
0x7870
IOC Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCDATA	R/W	Executing the writing operation in this register will latch setup value into I/O PortC data register. Similarly, executing the read operation in this register will read the status from I/O PortC external pads.	Refer to the above table, I/O port configuration and function .

P_IOC_Buffer
0x7871
IOC Buffer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCBUF															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCBUF	R/W	Executing the read operation in this register will read the setup value from I/O PortC data register, which is previously latched by IOCDATA writing operation.	IOCBUF(R) =IOCDATA(W) =IOCBUF(W)

P_IOC_Dir
0x7872
IOC Direction Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOC DIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCDIR	R/W	This control register sets the direction of I/O PortC. In addition, the direction setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOC_Attrib 0x7873 IOC Attribution Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOCATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IOCATT	R/W	This control register defines the attribution of I/O PortC. In addition, the attribution setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOD_Data 0x7878 IOD Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODDATA	R/W	Executing the writing operation in this register will latch setup value into I/O PortD data register. Similarly, executing the read operation in this register will read the status from I/O PortD external pads.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOD_Buffer 0x7879 IOD Buffer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODBUF															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODBUF	R/W	Executing the read operation in this register will read the setup value from I/O PortD data register, which is previously latched by IODDATA writing operation.	IODBUF(R) =IODDATA(W) =IODBUF(W)

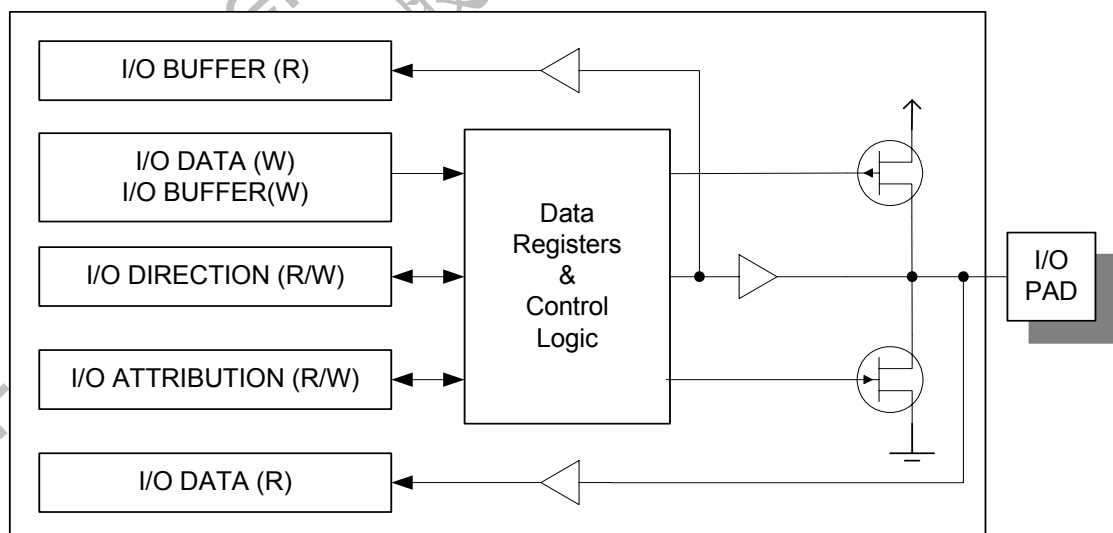
P_IOD_Dir	0x787A								IOD Direction Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODDIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODDIR	R/W	This control register sets the direction of I/O PortD. In addition, the direction setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

P_IOD_Attrib	0x787B								IOD Attribution Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IODATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	IODATT	R/W	This control register defines the attribution of I/O PortD. In addition, the attribution setup value can be read back from the same control register.	Refer to the above table, <i>I/O port configuration and function.</i>

5.5 I/O structure (diagrams)



5.6 Special Notes

Suppose a programmer intends to change the I/O configuration and finds out that two control registers need to be set up, the various setup sequence may have different outcomes. For example, changing the I/O mode from input with pull-low resistors to output with buffers requires changing both Direction and Attribution registers. Setting the Direction bit to "1" before Attribution bit to be "1" produces an unexpected short pulse. However, setting the Attribution bit to "1" before the Direction bit will not have the additional pulse.

5.7 Program Examples

```
R1 = 0xF0F0          // IOA[3..0] are input with pull-low resistor
[P_IOA_Data] = R1     // IOA[7..4] are input with pull high resistor
R1 = 0xFF00          // IOA[15..8] are all floating
[P_IOA_Attrib] = R1
R1 = 0x0000
[P_IOA_Dir] = R1
```

```
R1 = 0xF0F0          // IOC[3..0] are output buffer high (PMOS driven)
[P_IOC_Data] = R1     // IOC[7..4] are output buffer low (NMOS driven)
R1 = 0xFF00          // IOC[11..8] are output buffer low (NMOS driven)
[P_IOC_Attrib] = R1   // IOC[15..12] are output buffer high (PMOS driven)
R1 = 0xFFFF
[P_IOC_Dir] = R1
```

6 Interrupt

6.1 Introduction

GPL162002A/162003A provides many interrupt sources, which can also be wake-up sources. In other words, after system enters wait, halt, or sleep mode, an interrupt event will wake CPU up. For interrupt and wake-up capability from different modes, please refer to following table:

Interrupt Wakeup Capability Table for each mode

Interrupt / Wakeup Source	Wait Mode	Halt Mode	Sleep Mode
PortB[2:0] key change wakeup	O	O	O
EXTA wakeup	O	O	O
EXTB wakeup	O	O	O
DMA Transfer Complete Interrupt	O	X	X
USB Interrupt	O	X	X
I2C Transmit/Receive Interrupt	O	X	X
NAND Flash FIFO over/under flow interrupt	O	X	X
SD Controller Interrupt	O	X	X
Touch panel stylus tapped wakeup	O	O	O
UART/ IrDA receive wakeup	O	X	X
Serial Peripheral Interface (SPI) receive wakeup	O	X	X
LCD frame pulse (rising edge) wakeup	O	X	X
ADC Auto Sampling FIFO Full wakeup	O	X	X
TimerA wakeup	O	X	X
TimerB wakeup	O	X	X
TimerC wakeup	O	X	X
TimerD wakeup	O	X	X
Key Scan wakeup	O	X	X
TimebaseA wakeup	O	O	X
TimebaseB wakeup	O	O	X
TimebaseC wakeup	O	O	X
HMS (Hour/Minute/Second) wakeup	O	O	X
Alarm wakeup	O	O	X
Scheduler wakeup	O	O	X
Low voltage detect wakeup	O	O	X
Audio Channel A FIFO Empty wakeup	O	X	X
Audio Channel B FIFO Empty wakeup	O	X	X
TFT Under Flow Error wakeup (note1)	O	X	X
TFT Frame End wakeup (note 1)	O	X	X
ADC Conversion Ready wakeup	O	X	X

O: Supported, X: Not Supported

Note1: GPL162003 without TFT control interrupt wake up source.

All interrupts are level-triggered. That is, an interrupt flag has to be cleared when interrupt service begins; otherwise, CPU will re-enter the Interrupt service routine again. Most interrupt flags depicted in this chapter are read-only (reference-only). To enable or clear these interrupts, programmers must write corresponding control bits individually, which are depicted in the following chapter for each module (peripheral). The exceptions are the key change interrupt and external interrupts (EXTA and EXTB).

In addition, there are also interrupt flags which have the same function with these read-only interrupt flags, depicted in this chapter. For example, Reading from P_TimerA_Ctrl.bit15 is the same with reading from P_INT_Status2.bit12.

Generalplus suggests programmers do not use TimbaseA/B/C and scheduler as halt/sleep mode wake-up sources because the TimebaseA/B/C and scheduler interrupts occurs more quickly than the time that CPU wakes up from halt/sleep mode. As a result, the TimebaseA/B/C and scheduler interrupt flags will not be held from halt/sleep wake up.

6.2 Peripheral Interrupt Arrangement

As depicted in **Section: Memory, Vectors**, there are 11 interrupt events on Sunplus 16-bit CPU (μnSP^{\circledR}): Software Break, Fast Interrupt (FIQ), Reset and IRQ [7...0]. GPL162002A/162003A peripheral interrupts are distributed on nine of above interrupt sources, FIQ and IRQ [7...0]. The following table depicts the peripheral interrupt arrangement. Note that some peripheral interrupts can be configured as FIQ or as one of IRQ; refer to P_INT_Priority1 and P_INT_Priority2 control registers for details.

Interrupt Type	Possible Peripheral Interrupt	Flag Register
FIQ	Key Change Interrupt TFT Under Flow Error Interrupt ⁷ TFT Frame End Interrupt ⁷ UART/ IrDA Interrupt Serial Peripheral Interface (SPI) Interrupt LCD Frame Pulse Rising Edge (FP) Interrupt Touch Panel Interrupt (Stylus Tapped Interrupt) ADC Auto Sampling FIFO Full Interrupt AD Conversion Ready Interrupt Audio Channel A FIFO Empty Interrupt Audio Channel B FIFO Empty Interrupt External A Interrupt (rising or falling edge of IOD12) External B Interrupt (rising or falling edge of IOD13) TimerA Up-Counter Overflow or Capture or Comparison event Interrupt TimerB Up-Counter Overflow or Capture or Comparison event Interrupt TimerC Up-Counter Overflow or Capture or Comparison event Interrupt TimerD Up-Counter Overflow DMA Transfer Interrupt USB Interrupt I2C Transmit/Receive Interrupt NAND Flash FIFO over/under flow interrupt SD Controller Interrupt Key-Scan Interrupt	P_INT_Status1 P_INT_Status2
IRQ0	Audio Channel A FIFO Empty Interrupt Audio Channel B FIFO Empty Interrupt	P_INT_Status1
IRQ1	ADC Auto Sampling FIFO Full Interrupt AD Conversion Ready Interrupt ⁶ Touch Panel Interrupt (Stylus Tapped Interrupt)	P_INT_Status1
IRQ2	External A Interrupt (rising or falling edge of IOD12) External B Interrupt (rising or falling edge of IOD13)	P_INT_Status1

Interrupt Type	Possible Peripheral Interrupt	Flag Register
IRQ3	UART/ IrDA Interrupt ¹ Serial Peripheral Interface (SPI) Interrupt DMA Transfer Interrupt ⁴ USB Interrupt ⁵	P_INT_Status1
IRQ4	TimerA Up-Counter Overflow or Capture or Comparison event Interrupt ² TimerB Up-Counter Overflow or Capture or Comparison event Interrupt ² TimerC Up-Counter Overflow or Capture or Comparison event Interrupt ² TimerD Up-Counter Overflow	P_INT_Status2
IRQ5	Key Change Interrupt LCD Frame Pulse Rising Edge (FP) Interrupt TFT Under Flow Error Interrupt ⁷ TFT Frame End Interrupt ⁷ SP Serial Bus Interrupt	P_INT_Status1
	SD Controller Interrupt I2C Transmit/Receive Interrupt NAND Flash FIFO over/under flow interrupt	P_INT_Status2
IRQ6	Key-Scan Interrupt TimeBaseC Interrupt Scheduler Interrupt	P_INT_Status2
IRQ7	TimeBaseA Interrupt TimeBaseB Interrupt Alarm Interrupt Hour / Minute / Second / Half-Second Interrupt ³	P_INT_Status2

- There are three interrupt events in UART/IrDA module: reception, transmission and reception timeout interrupts. Any one of these three interrupt sources can trigger FIQ or IRQ3. Therefore, in FIQ or IRQ3 interrupt service routine, programmers should read P_UARTIrDA_Ctrl.bit [15...13] to distinguish which interrupt event happens. Note that 3-bit content reading from P_UARTIrDA_Ctrl.bit [15...13] sometimes have more than one "1". That means more than one interrupt event occur. Please refer to **chapter: UART/IrDA Interface**.
- There are three interrupt events in TimerA, TimerB, and TimerC module: up-counter overflow event, capture event and comparison event. One of above interrupt sources can trigger FIQ or IRQ4. Therefore, in FIQ or IRQ4 interrupt service routine, programmers should read P_TimerA_CCP_Ctrl.bit [15:14], P_TimerB_CCP_Ctrl.bit [15:14], P_TimerC_CCP_Ctrl.bit [15:14], to distinguish which interrupt event takes place. Note that, unlike UART / IrDA interrupt, only one of above three events will occur at one time. Please refer to **chapter: Timer/Counter**.

3. There are four interrupt events in Hour / Minute / Second / Half-Second Interrupt: Hour, Minute, Second, and Half-Second Interrupt. Any one of these four interrupt sources can trigger IRQ7. Therefore, in IRQ7 interrupt service routine, programmers should read P_RTC_INT_Status [3...0] to distinguish which interrupt event is generated. Note that 4-bit content reading from P_RTC_INT_Status [3...0] sometimes shows more than one interrupt event happening. For example, reading from P_RTC_INT_Status [3...0] will be "1111" when clock register is from 15:59:59 to 16:0:0. Please refer to **chapter: Real Time Clock**.
4. There are four interrupt sources, DMA1~DMA3, in DMA Transfer Interrupt. Each DMA source includes two interrupt events that a transfer is finished and timeout. Any one of these interrupt sources can trigger IRQ3. Therefore, in IRQ3 DMA interrupt service routine, programmers should read P_DMA_INT to distinguish which interrupt event is generated. Please refer to **chapter: DMA AND BRIDGE Controller**.
5. There are several interrupt events in USB function, for details, please refer to **chapter: USB Controller**.
6. There are three interrupt events in AD Conversion Ready Interrupt. These are manual mode AD convert ready, line-in right channel overflow and line-in left channel or mic-in overflow. Any one of these interrupt sources can trigger IRQ1. Therefore, in IRQ1 AD Conversion Ready interrupt service routine, programmers should read P_MADC_Ctrl, P_HQADC_R_Gain, and P_HQADC_L_Gain to distinguish which interrupt event is generated. Please refer to **chapter: Analog Input**.
7. On GPL162003 body, the FIQ and IRQ TFT interrupt vector are invalidly.

6.3 Control Registers

Global Interrupt Control Register Summary Table

Name	Address	Description
P_INT_Status1	0x78A0	Interrupt Status Register 1
P_INT_Status2	0x78A1	Interrupt Status Register 2
P_INT_Priority1	0x78A4	Interrupt Priority Register 1
P_INT_Priority2	0x78A5	Interrupt Priority Register 2
P_MINT_Ctrl	0x78A8	Miscellaneous Interrupt Control Register

P_INT_Status1		0x78A0										Interrupt Status 1 Register					
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		KEYIF	ADCRIF	TFTUFIF	TFTFEIF	UTIRIF	SPIIF	FPIF	TPIF	ASIF	-	AUDBIF	AUDAIF	USB	DMA	EXTBIF	EXTAIF
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	KEYIF	R/W	Key-change Interrupt status. Write "1" to clear the flag. Programmers need to read P_MINT_Ctrl to determine which key is changing. This bit is set to "1" by hardware if the key change interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	ADCRIF	R	AD Conversion Ready Interrupt status. This bit is set to "1" by hardware if the AD Conversion interrupt is asserted. For details, refer to chapter: Analog Input interface .	0= Not Occurred 1= Occurred
13	TFTFEIF	R	TFT Under Flow Interrupt status. This bit is set to "1" by hardware if the TFT under flow error interrupt is asserted. For details, refer to chapter: TFT LCD interface . This bit is invalid on GPL162003.	0= Not Occurred 1= Occurred
12	TFTFEIF	R	TFT Frame End Interrupt status. This bit is set to "1" by hardware if the TFT frame end interrupt is asserted. For details, refer to chapter: TFT LCD interface . This bit is invalid on GPL162003.	0= Not Occurred 1= Occurred

Bit	Function	Type	Description	Condition
11	UTIRIF	R	UART/IrDA Interrupt status. This bit is set to “1” by hardware if the UART/IrDA interrupt is asserted. For details, refer to chapter: UART/IrDA interface.	0= Not Occurred 1= Occurred
10	SPIIF	R	Serial Peripheral Interface (SPI) Interrupt status. This bit is set to “1” by hardware if the SPI interrupt is asserted. For details, refer to chapter: Serial Peripheral interface.	0= Not Occurred 1= Occurred
9	FPIF	R	FP Interrupt status. This bit is set to “1” by hardware if the LCD-FP-signal rising edge interrupt is asserted. For detail, refer to chapter: LCD interface.	0= Not Occurred 1= Occurred
8	TPIF	R	Touch Panel (Stylus Tapped) Interrupt status. This bit is set to “1” by hardware if the touch panel interrupt is asserted. For details, refer to chapter: Analog Input interface.	0= Not Occurred 1= Occurred
7	ASIF	R	ADC Auto Sample Mode FIFO Full Interrupt status. This bit is set to “1” by hardware if the Auto Sample FIFO Full (included microphone channel) interrupt is asserted. For details, refer to chapter: Analog Input interface.	0= Not Occurred 1= Occurred
6			Reserved	
5	AUDBIF	R	Audio Channel B FIFO Empty Interrupt status. This bit is set to “1” by hardware if the audio channel B FIFO empty interrupt is asserted. For details, refer to chapter: Audio Output interface.	0= Not Occurred 1= Occurred
4	AUDAIF	R	Audio Channel A FIFO Empty Interrupt status. This bit is set to “1” by hardware if the audio channel A FIFO empty interrupt is asserted. For details, refer to chapter: Audio Output interface.	0= Not Occurred 1= Occurred
3	USB	R	USB Interrupt. For details, refer to chapter: USB interface.	0= Not Occurred 1= Occurred
2	DMA	R	DMA transfer complete interrupt. This bit is set to “1” by hardware if the one of the DMATCR of DMA channels reaches 0 and the	0= Not Occurred 1= Occurred

Bit	Function	Type	Description	Condition
			corresponding DMA interrupt is enabled. For details, refer to chapter: DMA Controller .	
1	EXTBIF	R/W	External Interrupt B status. Write '1' to clear the flag. This bit is set to "1" by hardware if the external interrupt B is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
0	EXTAIF	R/W	External Interrupt A status. Write '1' to clear the flag. This bit is set to "1" by hardware if the external interrupt A is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag

P_INT_Status2

0x78A1

Interrupt Status 2 Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDIF	TMCIF	TMBIF	TMAIF	KSIF	TMBCIF	TMBBIF	TMBAIF	-	SD	I2C	NAND	-	SCHIF	ALMIF	HMSIF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMDIF	R	TimerD Interrupt Flag. This bit is set to "1" by hardware if the TimerD Up-counter overflow or Capture or Comparison event interrupt is asserted. For details, refer to chapter: Timer/Counter interface .	0= Not Occurred 1= Occurred
14	TMCIF	R	TimerC Interrupt Flag. This bit is set to "1" by hardware if the TimerC Up-counter overflow or Capture or Comparison event interrupt is asserted. For details, refer to chapter: Timer/Counter interface .	0= Not Occurred 1= Occurred
13	TMBIF	R	TimerB Interrupt Flag. This bit is set to "1" by hardware if the TimerB Up-counter overflow or Capture or Comparison event interrupt is asserted. For details, refer to chapter: Timer/Counter interface .	0= Not Occurred 1= Occurred
12	TMAIF	R	TimerA Interrupt Flag. This bit is set to "1" by hardware if the TimerA Up-counter overflow or Capture or Comparison event interrupt is asserted. For details, refer to chapter: Timer/Counter interface .	0= Not Occurred 1= Occurred
11	KSIF	R	Key-Scan Interrupt. This bit is set to "1" by hardware if the Key-Scan is finished. For details, refer to chapter: Key Scan interface .	0= Not Occurred 1= Occurred

Bit	Function	Type	Description	Condition
10	TMBCIF	R	TimeBaseC Interrupt Flag. This bit is set to “1” by hardware if the TimebaseC interrupt is asserted. For details, refer to chapter: TimeBase interface .	0= Not Occurred 1= Occurred
9	TMBBIF	R	TimeBaseB Interrupt Flag. This bit is set to “1” by hardware if the TimebaseB interrupt is asserted. For details, refer to chapter: TimeBase interface .	0= Not Occurred 1= Occurred
8	TMBAIF	R	TimeBaseA Interrupt Flag. This bit is set to “1” by hardware if the TimebaseA interrupt is asserted. For details, refer to chapter: TimeBase interface .	0= Not Occurred 1= Occurred
7			Reserved	
6	SD	R	SD Controller Interrupt. This bit is set to “1” by hardware if one of the events of SD controller occurs. For details, refer to chapter: SD/MMC interface .	0= Not Occurred 1= Occurred
5	I2C	R	I2C Controller Interrupt. This bit is set to “1” by hardware if one byte is transmitted/ received on the I2C bus or when address matches in I2C Slave mode. For details, refer to chapter: I2C interface .	0= Not Occurred 1= Occurred
4	NAND	R	NAND Flash Controller Interrupt. For details, refer to chapter: NAND Flash interface .	0= Not Occurred 1= Occurred
3			Reserved	
2	SCHIF	R	Schedule Interrupt Flag. This bit is set to “1” by hardware if the scheduler interrupt is asserted. For details, refer to chapter: Real Time Clock .	0= Not Occurred 1= Occurred
1	ALMIF	R	Alarm Interrupt Flag. This bit is set to “1” by hardware if the alarm interrupt is asserted. For details, refer to chapter: Real Time Clock .	0= Not Occurred 1= Occurred
0	HMSIF	R	HMS Interrupt Flag. This bit is set to “1” by hardware if the hour or minute or second or half-second interrupt is asserted. For details, refer to chapter: Real Time Clock .	0= Not Occurred 1= Occurred

P_INT_Priority1
0x78A4
Interrupt Priority 1 Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	KEYIP	ADCRIP	TFTUFIP	TFTFEIP	UTIRIP	SPIIP	FPIP	TPIP	ASPIP	-	AUDBIP	AUDAIP	USBIP	DMAIP	EXTBIP	EXTAIP
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	KEYIP	R/W	Key-change interrupts priority.	0= IRQ5 1= FIQ
14	ADCRIP	R/W	AD Conversion Ready interrupts priority.	0= IRQ1 1= FIQ
13	TFTUFIP	R/W	TFT Under Flow Error interrupts priority. This bit is invalid on GPL162003.	0= IRQ5 1= FIQ
12	TFTFEIP	R/W	TFT Frame End interrupts priority. This bit is invalid on GPL162003.	0= IRQ5 1= FIQ
11	UTIRIP	R/W	UART/ IrDA interrupt priority.	0= IRQ3 1= FIQ
10	SPIIP	R/W	Serial Peripheral Interface (SPI) interrupts priority.	0= IRQ3 1= FIQ
9	FPIP	R/W	LCD Frame Pulse Signal Rising Edge(FP) interrupt priority.	0= IRQ5 1= FIQ
8	TPIP	R/W	Touch panel (Stylus Tapped) interrupt priority.	0= IRQ1 1= FIQ
7	ASPIP	R/W	ADC Auto Sampling FIFO Full interrupts priority.	0= IRQ1 1= FIQ
6			Reserved	
5	AUDBIP	R/W	Audio Channel B FIFO Empty Interrupt priority.	0= IRQ0 1= FIQ
4	AUDAIP	R/W	Audio Channel A FIFO Empty Interrupt priority.	0= IRQ0 1= FIQ
3	USBIP	R/W	USB Interrupt priority.	0= IRQ3 1= FIQ
2	DMAIP	R/W	DMA transfer complete Interrupt priority.	0= IRQ3 1= FIQ
1	EXTBIP	R/W	External Interrupt B priority.	0= IRQ2 1= FIQ
0	EXTAIP	R/W	External Interrupt A priority.	0= IRQ2 1= FIQ

P_INT_Priority2
0x78A5
Interrupt Priority 2 Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDIP	TMCIP	TMBIP	TMAIP	KSIP	-	-	-	-	SD	I2C	NAND	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
15	TMDIP	R/W	TimerD up-counter or capture or comparison event overflow interrupt priority.	0= IRQ4 1= FIQ

Bit	Function	Type	Description	Condition
14	TMCIP	R/W	TimerC up-counter or capture or comparison event overflow interrupt priority.	0= IRQ4 1= FIQ
13	TMBIP	R/W	TimerB up-counter or capture or comparison event overflow interrupt priority.	0= IRQ4 1= FIQ
12	TMAIP	R/W	TimerA up-counter or capture or comparison event overflow interrupt priority.	0= IRQ4 1= FIQ
11	KSIP	R/W	Key Scan Interrupt Priority.	0= IRQ6 1= FIQ
[10:7]			Reserved	
6	SD	R/W	SD Controller Interrupt priority.	0= IRQ5 1= FIQ
5	I2C	R/W	I2C Controller Interrupt priority.	0= IRQ5 1= FIQ
4	NAND	R/W	NAND Flash Controller Interrupt priority.	0= IRQ5 1= FIQ
[3:0]			Reserved	

P_MINT_Ctrl
0x78A8
Miscellaneous Interrupt Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	KC2IF	KC2EN	KC1IF	KC1EN	KC0IF	KC0EN	-	-	-	-	EXTBIS	EXTAIS	-	EXTBEN	EXTAEN	
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
15	KC2IF	R/W	Key-change 2 Interrupt Flag. If this bit is set to "1", IOB2 key change Interrupt happens.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
14	KC2EN	R/W	Key-change 2 Interrupt Enable. If this bit is set to "1" and bit9 (KCEN) in P_Clock_Ctrl is set to 1, IOB2 key change Interrupt is enabled.	0= Disabled 1= Enabled
13	KC1IF	R/W	Key-change 1 Interrupt Flag. If this bit is set to "1", IOB1 key change Interrupt happens.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
12	KC1EN	R/W	Key-change 1 Interrupt Enable. If this bit is set to "1" and bit9 (KCEN) in P_Clock_Ctrl is set to 1, IOB1 key change Interrupt is enabled.	0= Disabled 1= Enabled
11	KC0IF	R/W	Key-change 0 Interrupt Flag. If this bit is set to "1", IOB0 key change	Read 0= Not Occurred Read 1= Occurred

Bit	Function	Type	Description	Condition
			Interrupt happens.	Write 0= No Effect Write 1= Clear the flag
10	KC0EN	R/W	Key-change 0 Interrupt Enable. If this bit is set to "1" and bit9 (KCEN) in P_Clock_Ctrl is set to 1, IOB0 key change Interrupt is enabled.	0= Disabled 1= Enabled
[9:6]			Reserved	
5	EXTBIS	R/W	EXTB Interrupt Edge Selection. This bit is valid only when EXTBIEEN control bit is set to "1". If this bit is set to "1", the EXTB is triggered on rising edge. Otherwise, it is triggered on falling edge.	0= Falling edge triggered 1= Rising edge triggered
4	EXTAIS	R/W	EXTA Interrupt Edge Selection. This bit is valid only when EXTAIEEN control bit is set to "1". If this bit is set to "1", the EXTA is triggered on rising edge. Otherwise, it is triggered on falling edge.	0= Falling edge triggered 1= Rising edge triggered
[3:2]			Reserved	
1	EXTBIEN	R/W	EXTB interrupt enable. If this bit is set as "1", PortD13 will be configured as external interrupt B input pin. If this bit is cleared to "0", PortD13 remains as GPIO.	0= Disabled 1= Enabled
0	EXTAIEN	R/W	EXTA interrupt enable. If this bit is set as "1", PortD12 will be configured as external interrupt A input pin. If this bit is cleared to "0", PortD12 remains as GPIO.	0= Disabled 1= Enabled

6.4 Program Examples

```
.PUBLIC          _BREAK, _FIQ, _IRQ6;

//*****

_BREAK:          // Software Break ISR
                 reti

_FIQ:
                 push r1 to [sp]          // Save destroyed CPU register r1 to stack
                 r1 = [P_INT_Status1]
                 r1 = r1 & C_INT_AUDAFIFOEmpty
                 jz  L_End_AudioCHA_ISR?
                 [P_CHA_Ctrl] = r1
                 r1 = [P_CHA_Ctrl]        // Clear Interrupt Flag
                 .....
                 .....

L_End_AudioCHA_ISR?:
                 pop r1 from [sp]         // Restore original CPU register r1 from stack
                 reti

//*****

_IRQ6:
                 push r1, r3 to [sp]      // Save destroyed CPU register r1, r2, r3 to stack
                 r2 = [P_INT_Status2]
                 r1 = r2 & C_INT_TimeBaseC
                 jz  L_End_TimeBaseC_ISR?
                 r1 = [P_TimeBaseC_Ctrl]  // Clear Interrupt Flag
                 [P_TimeBaseC_Ctrl] = r1

L_End_TimeBaseC_ISR?:
                 .....
                 .....
                 pop r1, r3 from [sp]     // Restore original CPU register r1, r2, r3 from stack
                 reti
```

7 Timer/Counter

7.1 Timer Introduction

GPL162002A/162003A contains six 16-bit timers/counters: TimerA to TimerF. TimerA, TimerB, and TimerC support Capture/Comparison/PWM (CCP) functions when cooperating with their own two 16-bit registers (a preload register and a CCP register). On the other hand, TimerD, TimerE, and TimerF have only one 16-bit preload register and therefore, these timers do not provide CCP functions. The clock sources of these six timers can be programmed by internal clock sources (timer mode) or from external I/O pin (counter mode). TimerE and TimerF do not support Interrupt function.

Main features

- Variance clock source selection for each timer source.
- Support two external clock sources.
- One external clock source provides 1/4, 1/16 pre-scalar.
- Each programmable clock source can be synchronous with CPU clock as for each timer source.
- Three timers support Capture function.
- Three timers support Comparison function.
- Three timers support Pulse Width Modulation (PWM) function.

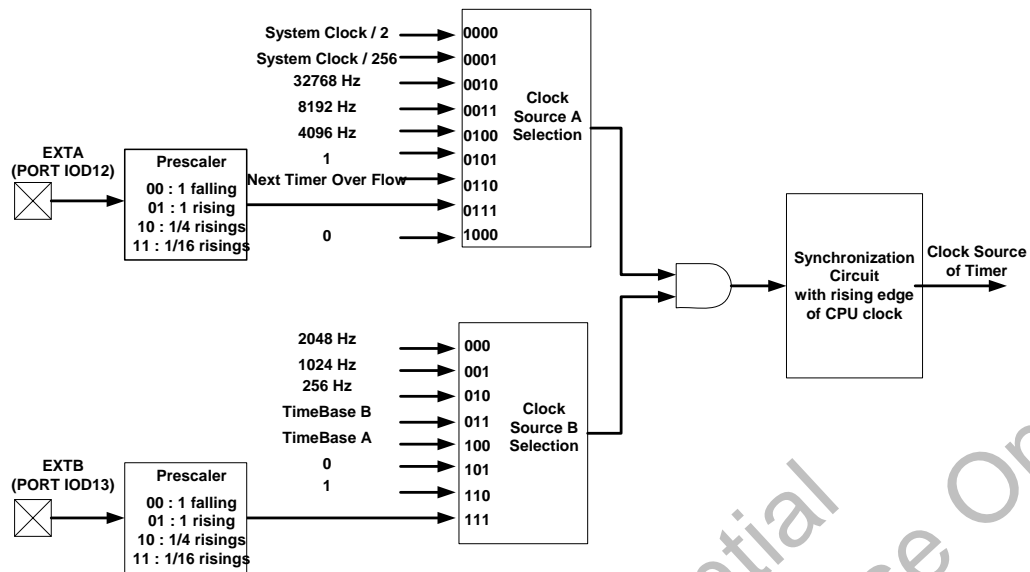
7.2 Timer Structure and Clock Source

There are two basic operation modes on GPL162002A/162003A timer/counter module - timer mode and counter mode.

In timer mode, the clock source is generated by internal clock sources such as CPUCLK/256, CPUCLK/2, 32768Hz, 8192Hz, 4096Hz, 1024Hz, 4Hz, 2Hz, 1Hz, or combinations (for detail, refer to clock source block diagram). Besides, each clock source can be synchronized with CPU clock in the way of setting the corresponding control bits.

In counter mode, the clock source is supported by external clock source pin (EXTA or EXTB). Additionally, signal from external clock source can be divided by 4 or 16 pre-scalar and will be capable of selecting trigger edge (rising or falling). To ensure the CPU clock synchronization procedure is completed, this external clock frequency has to be less than or equal to a half of CPUCLK.

Following are function diagrams of source clock selection of each timer.



Each of these 16-bit timers/counters has an up-counter and increment on the rising edge of internal clock source, or increment either on the rising or falling edge of external clock source. Initial value of the up-counter is stored in the pre-load register (preload register). When the timer/counter is enabled and overflow occurs, the initial value is loaded into counter on the next increment clock edge (synchronous load). At the same time, corresponding interrupt flag is set. If corresponding interrupt is enabled, it will issue an interrupt to CPU. For example, if the initial value is 0xFFFFC, the timer/counter will count by the sequence 0xFFFFC, 0xFFFFD, 0xFFFFE, 0xFFFFF, 0xFFFFC, 0xFFFFD, etc. In other words, the timer/counter's overflow frequency is (Source clock frequency) / (65536 - Preload value).

Formula:

Timer/Counter Overflow Interrupt Frequency = (Source clock frequency) / (65536 – Preload Register Value).

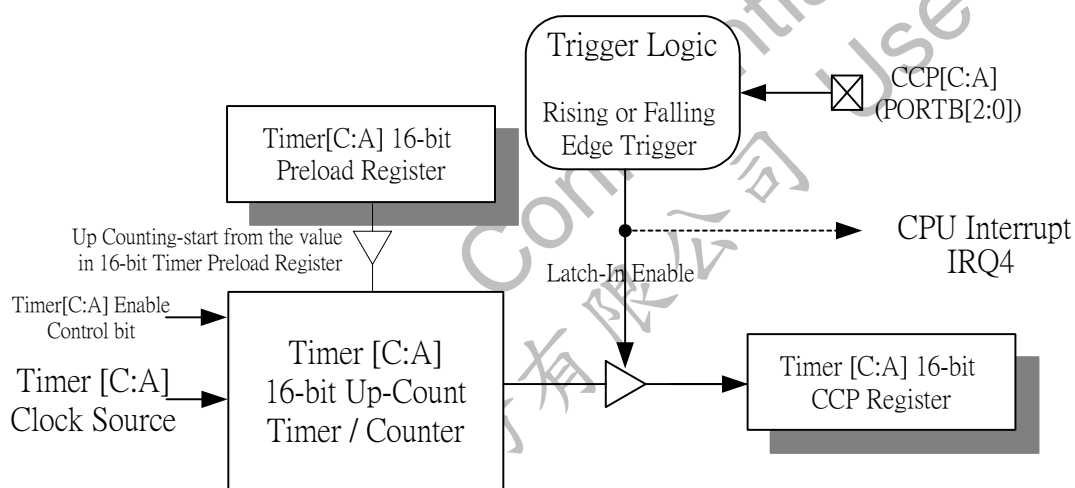
Note that if external Input A (EXTA/PortD12) is selected as counter clock source, this pin cannot be GPIO function. In other words, any GPIO setting on PortD12 will be in vain. Similarly, if external Input B (EXTB/PortD13) is selected as counter clock source, this pin cannot be GPIO function. That is, any GPIO setting on PortD13 will have no effect.

In addition, TimerA, TimerB, and TimerC offer Capture / Comparison / Pulse-Width-Modulation (CCP) special functions. If one of these three special functions is enabled, Control Register 2 and CCP Register of TimerA, TimerB, and TimerC should be set up appropriately. Note that TimerD to TimerF do not support CCP functions. The following three sections will depict the detailed operations for CCP function.

Capture Mode

In capture mode, the value in Timer/Counter register is latched in Capture/Comparison/PWM registers (CCP Register) at the selected edge (rising or falling) of external I/O pin (PortB [2:0]). The value in Timer/Counter Register can be latched every rising or falling edge of external I/O pin (refer to control register 2 for details). When a capture occurs, the interrupt flag is set and CPU is interrupted via IRQ4. The interrupt flag must be cleared by firmware.

Note that if another capture occurs before the value in the CCP Register is read, the old captured value will be lost. That is, the clock frequency from external I/O pin has to be at least one half of the timer/counter's clock source frequency. The corresponding GPIO pin, PortB [2:0], for the timer/counter is configured as INPUT pin automatically when TimerA, TimerB or TimerC is set as capture mode.

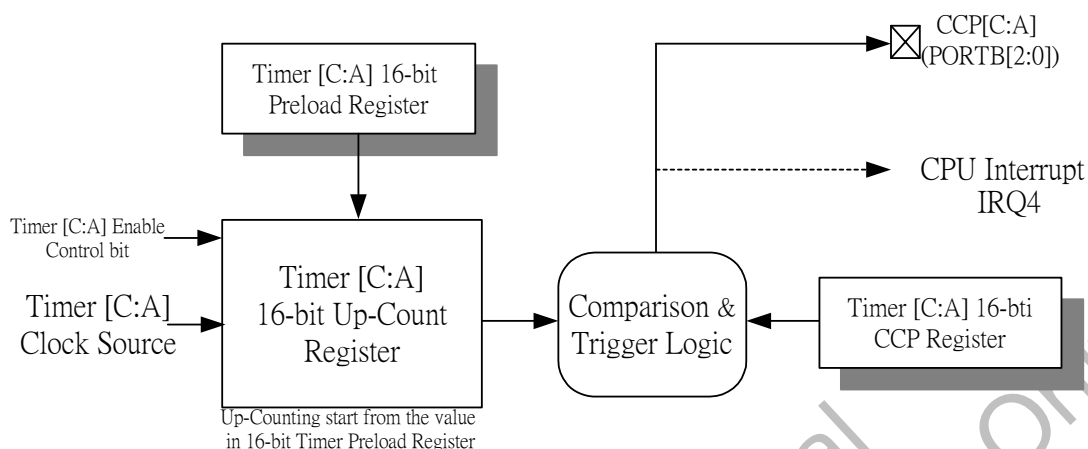


Timer [C:A] Capture Mode Function Diagram

When TimerA, TimerB, or TimerC is configured as capture mode (P_TimerX_CCP_Ctrl.bit [15, 14] = "01"), programmers can enable, disable, clear and read status of the corresponding capture event interrupt by accessing P_TimerX_Ctrl.bit [15.14]. At this time, timerA, TimerB, or timerC up-counter overflow interrupt will have no effect on any operation of P_TimerX_Ctrl.bit [15.14]. In other words, only one interrupt event is valid on the control bits (P_TimerX_Ctrl.bit [15.14]) at one time, either up-counter overflow interrupt or capture event interrupt.

Comparison mode

In comparison mode, the value to be compared with the 16-bit timer/counter registers is stored in CCP Register. The preloaded value is reloaded to a 16-bit timer/counter as well as the interrupt flag is set if they are matched. The corresponding PortB [2:0] I/O pin is configured as an OUTPUT pin automatically when comparison mode is set. Note that in this mode, the overflow of 16-bit timer/counter will reload the preloaded value.

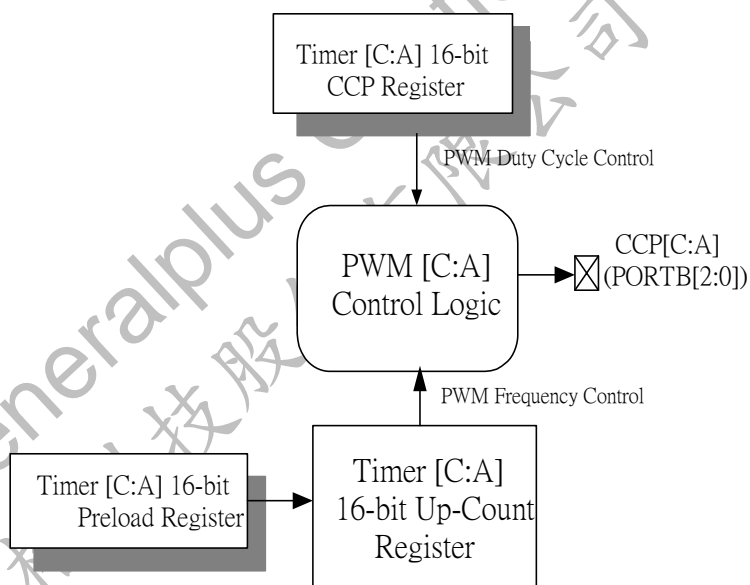
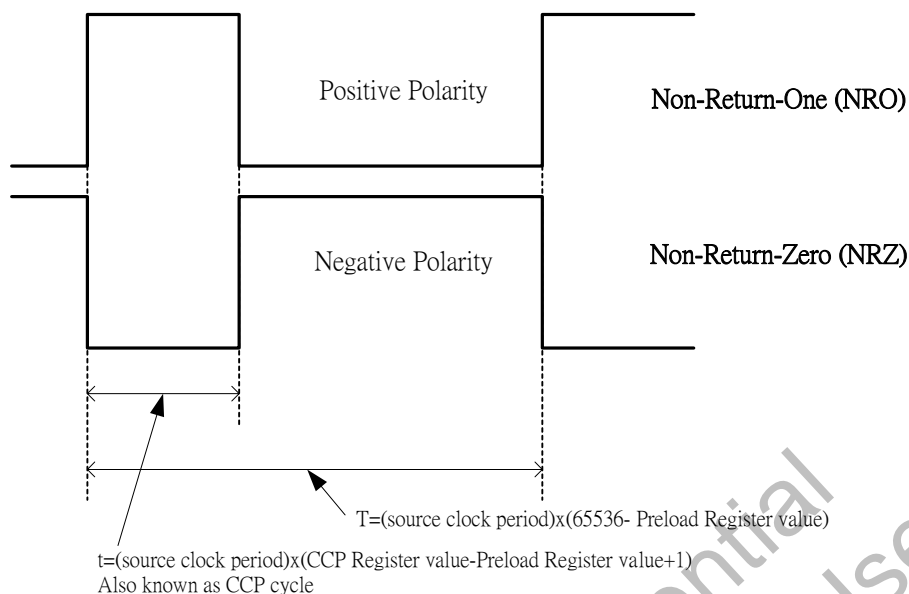


Timer [C:A] Comparison Mode Function Diagram

When TimerA or TimerB is configured as comparison mode (P_TimerX_CCP_Ctrl.bit [15:14] = "10"), programmers can enable, disable, clear and read status of the corresponding comparison event interrupt by accessing P_TimerX_Ctrl.bit[15.14]. At this time, timerA, TimerB, or timerC up-counter overflow interrupt will have no effect on any operation of P_TimerX_Ctrl.bit[15.14]. In other words, only one interrupt event is valid on the control bits (P_TimerX_Ctrl.bit[15.14]) at one time, either up-counter overflow interrupt or comparison event interrupt.

Pulse-Width-Modulation Mode

When a timer/counter is set as in Pulse-Width-Modulation (PWM) mode, the PortB [2:0] pin is configured as OUTPUT pin automatically. When the timer/counter is enabled, the PWM output will toggle at the overflow of timer/counter and at the end of CCP cycle. There are two output signal polarities on PWM output pin, positive and negative.



In Pulse-Width-Modulation (PWM) normal mode

PWM frequency is set by the preloaded register value. See the formula below:

$$\text{PWM Output Frequency} = (\text{Source Clock Frequency}) / (65536 - \text{Preload Register Value})$$

PWM duty cycle is set by the CCP register value. See the formula below:

$$\text{PWM Output Duty Cycle} = (\text{CCP Register} - \text{Preload_Register} + 1) / (65536 - \text{Preload Register})$$

$$\text{PWM period} = 1 / \text{PWM Frequency}$$

7.3 Control Registers

Timer Control Register Summary Table

Name	Address	Description
P_TimerA_Ctrl	0x78C0	TimerA Control Register
P_TimerA_CCP_Ctrl	0x78C1	TimerA Capture / Comparison / PWM Control Register
P_TimerA_Preload	0x78C2	TimerA Preload Register
P_TimerA_CCP_Reg	0x78C3	TimerA Capture / Comparison / PWM Register
P_TimerA_UpCount	0x78C4	TimerA up-count value
P_TimerB_Ctrl	0x78C8	TimerB Control Register
P_TimerB_CCP_Ctrl	0x78C9	TimerB Capture / Comparison / PWM Control Register
P_TimerB_Preload	0x78CA	TimerB Preload Register
P_TimerB_CCP_Reg	0x78CB	TimerB Capture / Comparison / PWM Register
P_TimerB_UpCount	0x78CC	TimerB up-count value
P_TimerC_Ctrl	0x78D0	TimerC Control Register
P_TimerC_CCP_Ctrl	0x78D1	TimerC Capture / Comparison / PWM Control Register
P_TimerC_Preload	0x78D2	TimerC Preload Register
P_TimerC_CCP_Reg	0x78D3	TimerC Capture / Comparison / PWM Register
P_TimerC_UpCount	0x78D4	TimerC up-count value
P_TimerD_Ctrl	0x78D8	TimerD Control Register
P_TimerD_Preload	0x78DA	TimerD Preload Register
P_TimerD_UpCount	0x78DC	TimerD up-count value
P_TimerE_Ctrl	0x79C0	TimerE Control Register
P_TimerE_Preload	0x79C2	TimerE Preload Register
P_TimerE_UpCount	0x79C4	TimerE up-count value
P_TimerF_Ctrl	0x79C8	TimerF Control Register
P_TimerF_Preload	0x79CA	TimerF Preload Register
P_TimerF_UpCount	0x79CC	TimerF up-count value

P_TimerA_Ctrl				0x78C0				TimerA Control Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	TMAIF/C	TMAIE	TMAEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_TimerB_Ctrl				0x78C8				TimerB Control Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	TMBIF/C	TMBIE	TMBEN	-	EXTASEL		EXTBSEL		-	SRCBSEL			SRCASEL						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_TimerC_Ctrl		0x78D0								TimerC Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TMCIF/C	TMCIE	TMCEN	-	EXTASEL	EXTBSEL	-		SRCBSEL				SRCASEL			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD_Ctrl		0x78D8								TimerD Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TMDIF/C	TMDIE	TMDEN	-	EXTASEL	EXTBSEL	-		SRCBSEL				SRCASEL			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerE_Ctrl		0x79C0								TimerE Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TMEIF/C	TMEIE	TMEEN	-	EXTASEL	EXTBSEL	-		SRCBSEL				SRCASEL			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerF_Ctrl		0x79C8								TimerF Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TMFIF/C	TMFIE	TMFEN	-	EXTASEL	EXTBSEL	-		SRCBSEL				SRCASEL			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMXIF/C	R/W	<p>TimerX Interrupt Flag.</p> <p>Write "1" to clear the flag.</p> <p>In timer/ counter mode, this bit is set when TimerX rolls over from FFFF and is cleared by writing this bit to "1".</p> <p>In capture mode, This bit is set when the external signal is driven, and timer/counter value is latched into TimerX CCP register.</p> <p>In comparison mode, this bit is set when the timer/counter value is the same as the value stored in TimerX CCP register.</p> <p>In PWM mode, reading or writing this bit has no effect.</p>	<p>Read 0= Not Occurred</p> <p>Read 1= Occurred</p> <p>Write 0 = No effect</p> <p>Write 1= Clear the flag</p>
14	TMXIE	R/W	<p>TimerX Interrupt Enable.</p> <p>If this bit is set to "1", and TimeX (Timer overflow or capture or compare event) interrupt happens, hardware will issue an IRQ4 or FIQ to CPU. If this bit is clear to "0", this interrupt will be masked.</p> <p>To select between IRQ4 and FIQ, please refer to Chapter Interrupt.</p>	<p>0= Disabled</p> <p>1= Enabled</p>

Bit	Function	Type	Description	Condition
			It should be noted the TimerE & TimerF will not issue interrupt to CPU.	
13	TMXEN	R/W	TimerX Enable If this bit is set to "1", TimerX will start to up count its 16-bit timer/counter register according the frequency of a selected clock source. If this bit is clear to "0", the TimerX will stop counting.	0= Disabled 1= Enabled
12			Reserved	
[11:10]	EXTASEL	R/W	External Input A (IOD12) pre-scalar setup. These two bits are used to select the pre-scalar function between external input signal and timer clock source.	00= EXT A every falling 01= EXT A every rising 10= EXT A every 4 risings 11= EXT A every 16 risings
[9:8]	EXTBSEL	R/W	External Input B (IOD13) pre-scalar setup. These two bits are used to select the pre-scalar function between external input signal and timer clock source.	00= EXTB every falling 01= EXTB every rising 10= EXTB every 4 risings 11= EXTB every 16 risings
7			Reserved	
[6:4]	SRCBSEL	R/W	Clock Source Group B selection. *Note: When TimebaseB or TimebaseA is selected, the corresponding timebase enable bit should be set or else the timer will not count. **Note: When external Input B ("111" in binary) is selected, PortD13 is configured as counter clock source. It cannot be GPIO function; that is, any GPIO setting on PortD13 will be in vain.	000= 2048Hz 001= 1024Hz 010= 256Hz 011= TimeBaseB* 100= TimeBaseA* 101= 0 (logic low) 110= 1 (logic high) 111= EXTB with Pre-scalar**
[3:0]	SRCASEL	R/W	Clock Source Group A selection. ***Note: Timer(X+1) and TimerX, form a 32-bit counter, and are configured as a cascade mode when Timer(X+1) overflow ("0110" in binary) is selected. ****Note: When external Input A ("0111" in binary) is selected, IOD12 is configured as counter clock source. It cannot be GPIO function; that is, any GPIO setting on IOD12 will be in vain.	0000= SYSCLK/2 0001= SYSCLK/256 0010= 32768Hz 0011= 8192Hz 0100= 4096Hz 0101= 1 0110= Timer(X+1) Overflow*** 0111= EXT A with Pre-scalar**** 1000= 0 1001~1111 = Reserved

P_TimerA CCP_Ctrl 0x78C1
TimerA CCP Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPAEN						CAPASEL					CMPASEL			PWMASEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB CCP_Ctrl 0x78C9
TimerB CCP Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPBEN						CAPBSEL					CMPBSEL			PWMBSEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC CCP_Ctrl 0x78D1
TimerC CCP Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPCEN						CAPCSEL					CMPCSEL			PWMCSEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]	CCPXEN	R/W	Operation Mode Selection. (Default is timer/counter mode) When capture mode is selected, PortBX becomes INPUT automatically and any GPIO setting will have no effect. When comparison and PWM mode is selected, PortBX becomes OUTPUT automatically; any GPIO setting will have no effect. It should be noted TimerD, TimerE & TimerF do not have their specified output pin, so capture and PWM mode is not valid for these four timer.	00= CCP Mode Disabled 01= Capture Enabled 10= Comparison Enabled 11= PWM Enabled
[13:10]			Reserved	
[9:8]	CAPXSEL	R/W	Capture Operation mode Selection. These 2 bits are valid only when CCPXEN is set to "01" in binary.	00= every falling 01= every rising 10= reserved 11= reserved
[7:6]			Reserved	
[5:4]	CMPXSEL	R/W	Comparison Operation Mode Selection. These 2 bits are valid only when CCPXEN is set to "10" in binary.	00= high pulse on CCP[X] 01= low pulse on CCP[X] 10= unaffected on CCP[X] 11= reserved
[3:2]			Reserved	
[1:0]	PWMXSEL	R/W	PWM Operation Mode Selection. These 2 bits are valid only when CCPXEN is set to "11" in binary.	00= PWM mode, NRO output 01= PWM mode, NRZ output 10= reserved

Bit	Function	Type	Description	Condition
				11= reserved

P_TimerA_Preload		0x78C2				TimerA Preload Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB_Preload		0x78CA				TimerB Preload Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TMBPLR																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TimerC_Preload		0x78D2				TimerC Preload Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD_Preload				0x78DA				TimerD Preload Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TMDPLR																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TimerE_Preload		0x79C2					TimerE Preload Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMEPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerF_Preload	0x79CA																TimerF Preload Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Function	TMFPLR																							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								

Bit	Function	Type	Description	Condition
[15:0]	TMXPLR	R/W	TimerX pre-load register.	

P_TimerA CCP_Reg	0x78C3				TimerA Capture / Comparison / PWM Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMACCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB_CCP_Reg	0x78CB						TimerB Capture / Comparison / PWM Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC	CCP_Reg	0x78D3					TimerC Capture / Comparison / PWM Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMXCCPR	R/W	TimerX Capture/Comparison/PWM register.	

P_TimerA UpCount	0x78C4							TimerA Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB UpCount	0x78CC						TimerB Up-Count									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC UpCount	0x78D4				TimerC Up-Count											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD UpCount	0x78DC				TimerD Up-Count											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerE UpCount	0x79C4							TimerE Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMEUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerF UpCount	0x79CC						TimerF Up-Count									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMFUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TMXUCR	R	TimerX up-count. The up-counter's value can be read from this register.	

7.4 Program Examples

```

.DEFINE      SYSCLK      48000000

//*****
L_TimerCClockSourcePolling:
    r1 = 0x0001
    [P_IOC_Attrib] = r1
    [P_IOC_Dir] = r1                //Setup IOC0 as output buffer low

    Int off
    r1 = (0x10000-SYSCLK/2/8000)    //Setup TimerC preload value
    [P_TimerC_Preload] = r1        //so that up-counter overflow frequency = 8000Hz
    r1 = 0x2000
    [P_TimerC_Ctrl] = r1            //Enable TimerC

L_CheckTimerCOverflow?:
    r1 = [P_TimerC_Ctrl]
    jpl L_CheckTimerCOverflow?
    [P_TimerC_Ctrl] = r1            //Clear TimerC up-counter-overflow interrupt flag
    r1 = [P_IOC_Buffer]            //Read Previous PortC Setup
    r1 = r1 xor 0x0001
    [P_IOC_Data] = r1              //Toggle PortC0 for 50% duty square wave
    jmp L_CheckTimerCOverflow?    // and frequency is 4000 Hz (8000/2)

//*****
L_TimerAPWMPolling:
    r1 = 0x1000                    //IOB0 will output a square wave with 1/16 duty
    [P_TimerA_CCP_Reg] = r1        //cycle and frequency is CPUCLK/2/65536
    r1 = 0x0000
    [P_TimerA_Preload] = r1
    r1 = 0xC000
    [TimerA_CCP_Ctrl] = r1
    r1 = 0xA060
    [P_TimerA_Ctrl] = r1
    jmp $

```

8 Timebase

8.1 Introduction

A timebase, generated from 32768Hz source, is a combination of frequency selections. GPL162002A/162003A supports three timebases, accompanied with their interrupt mechanism; these facilitate timing control for most of projects. In additions, a timebase also provides variety of frequency selections to clock source of Timer.

These three timebases are,

- 1Hz/ 2Hz/ 4Hz frequency programmable for TimebaseA.
- 8Hz/ 16Hz/ 32Hz/ 64Hz frequency programmable for TimebaseB.
- 128Hz/ 256Hz/ 512Hz/ 1024Hz frequency programmable for TimebaseC.

8.2 Timebase structure and clock source



8.3 Control Registers

Timebase Control Register Summary Table

Name	Address	Description
P_TimeBaseA_Ctrl	0x78B0	TimeBaseA Control Register
P_TimeBaseB_Ctrl	0x78B1	TimeBaseB Control Register
P_TimeBaseC_Ctrl	0x78B2	TimeBaseC Control Register
P_TimeBase_Reset	0x78B8	TimeBase Counter Reset Register

P_TimeBaseA_Ctrl			0x78B0 TimeBaseA Control Register																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	TMBAIF/C	TMBAIE	TMBASEN	-	-	-	-	-	-	-	-	-	-	-	-	TMBAS			
Default	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1			

Bit	Function	Type	Description	Condition
15	TMBAIF/C	R/W	TimebaseA interrupt Flag. Write "1" to clear the flag. If TimebaseA interrupt occurs, this flag is set to "1" by hardware.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
14	TMBAIE	R/W	TimebaseA Interrupt Enable. If this bit is set to “1” and TimeBaseA interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked. Generalplus suggests programmers do not use TimbaseA as halt/sleep mode wake-up source because the TimebaseA interrupt occurs more quickly than the time CPU wakes up from halt/sleep mode. As a result, the TimebaseA interrupt flag will not be held from halt/sleep wake-up.	0= Disabled 1= Enabled
13	TMBAEN	R/W	TimeBaseA Module Enable. If this bit is set to “1”, TimeBaseA module will be enabled; on the contrary, it will be disabled for power consumption consideration.	0= Disabled 1= Enabled
[12:2]			Reserved	
[1:0]	TMBAS	R/W	TimebaseA frequency selection. There are three frequency sources on TimebaseA, 1Hz, 2Hz, and 4Hz. These two control bits are to select one of three frequencies.	00= reserved 01= 1Hz 10=2Hz 11= 4Hz

P_TimeBaseB_Ctrl
0x78B1
TimeBaseB Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBBIF/C	TMBBIE	TMBBEN	-	-	-	-	-	-	-	-	-	-	-	TMBBS	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TMBBIF/C	R/W	TimebaseB interrupt Flag. Write “1” to clear the flag. If TimebaseB interrupt occurs, this flag is set to “1” by hardware.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBBIE	R/W	TimebaseB Interrupt enable. If this bit is set to “1”, and TimeBaseB interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to “0”, this interrupt will be masked. Generalplus suggests programmers do not use TimbaseB as halt/sleep mode wake up	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			source because the TimebaseB interrupt occurs more quickly than the time CPU wakes up from halt/sleep mode. As a result, the TimebaseB interrupt flag will not be held from halt/sleep wake-up.	
13	TMBBEN	R/W	TimeBaseB Module Enable. If this bit is set to "1", TimeBaseB module will be enabled; on the contrary, it will be disabled for power consumption consideration.	0= Disabled 1= Enabled
[12:2]			Reserved	
[1:0]	TMBBS	R/W	TimebaseB Frequency Selection. There are four frequency sources on TimebaseB, 8Hz, 16Hz, 32Hz, and 64Hz. These two control bits are to select one of four frequencies.	00= 8Hz 01= 16Hz 10= 32Hz 11= 64Hz

P_TimeBaseC_Ctrl
0x78B2
TimeBaseC Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCIF/C	TMBCIE	TMBCEN	-	-	-	-	-	-	-	-	-	-	-	TMBCS	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*	0

Bit	Function	Type	Description	Condition
15	TMBCIF/C	R/W	TimebaseC Interrupt Flag. Write "1" to clear the flag. If TimebaseC interrupt occurs, this flag is set to "1" by hardware.	Read 0= Not Occurred Read 1= occurred Write 0 = No Effect Write 1= Clear the flag
14	TMBCIE	R/W	TimebaseC Interrupt enable. If this bit is set to "1" and TimeBaseC interrupt occurs, hardware will issue an IRQ6 to CPU. If this bit is cleared to "0", this interrupt will be masked. Generalplus suggests programmers do not use TimebaseC as halt/sleep mode wake up source because the TimebaseC interrupt occur occurs more quickly than the time that CPU wakes up from halt/sleep mode. As a result, the TimebaseC interrupt flag will not be held from halt/sleep wake-up.	0= Disabled 1= Enabled
13	TMBCEN	R/W	TimeBaseC Module Enable. If this bit is set to "1", TimeBaseC module will be enabled; on the contrary, it will be disabled for power consumption	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			consideration.	
[12:2]			Reserved	
[1:0]	TMBCS	R/W	TimebaseC Frequency Selection. There are four frequency sources on TimebaseC, 128Hz, 256Hz, 512Hz, and 1024Hz. These two control bits are to select one of four frequencies.	00= 128Hz 01= 256Hz 10= 512Hz 11= 1024Hz

P_TimeBase_Reset
0x78B8
TimeBase Reset Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCCR															
Default																

Bit	Function	Type	Description	Condition
[15:0]	TMBCCR	W	Writing 0x5555 to this control register will reset the internal timebase counter for precise timing correction/control.	

8.4 Program Examples

L_PollingTimeBase:

```

r1 = 0x0007
[P_IOA_Attrib] = r1
[P_IOA_Dir] = r1                                //Setup IOA[2..0] as output buffer low

r1 = 0xA001
[P_TimeBaseA_Ctrl] = r1                        // Setup TimeBaseA Frequency as 1Hz
[P_TimeBaseB_Ctrl] = r1                        // Setup TimeBaseB Frequency as 16Hz
[P_TimeBaseC_Ctrl] = r1                        // Setup TimeBaseC Frequency as 256Hz

```

L_CheckTimeBaseOverflow?:

```

r1=[ P_TimeBaseC_Ctrl]
jpl   L_EndCheckTimeBaseC?
[P_TimeBaseC_Ctrl]=r1                        //Clear TimeBaseC Flag
r1 = [P_IOA_Buffer]
r1 = r1 xor 0x0004
[P_IOA_Data] = r1                            // Toggle IOA2 (frequency = 0.5Hz = 1Hz/2 )

```

L_EndCheckTimeBaseC?:

```

r1=[ P_TimeBaseB_Ctrl]
jpl   L_EndCheckTimeBaseB?
[P_TimeBaseB_Ctrl]=r1                        //Clear TimeBaseB Flag
r1 = [P_IOA_Buffer]
r1 = r1 xor 0x0002
[P_IOA_Data] = r1                            // Toggle IOA1 (frequency = 8Hz = 16Hz/2 )

```

L_EndCheckTimeBaseB?:

```

r1=[ P_TimeBaseA_Ctrl]
jpl   L_CheckTimeBaseOverflow?
[P_TimeBaseA_Ctrl]=r1                        //Clear TimeBaseA Flag
r1 = [P_IOA_Buffer]
r1 = r1 xor 0x0001
[P_IOA_Data] = r1                            // Toggle IOA0 (frequency = 128Hz = 256Hz/2 )
jmp   L_CheckTimeBaseOverflow?

```

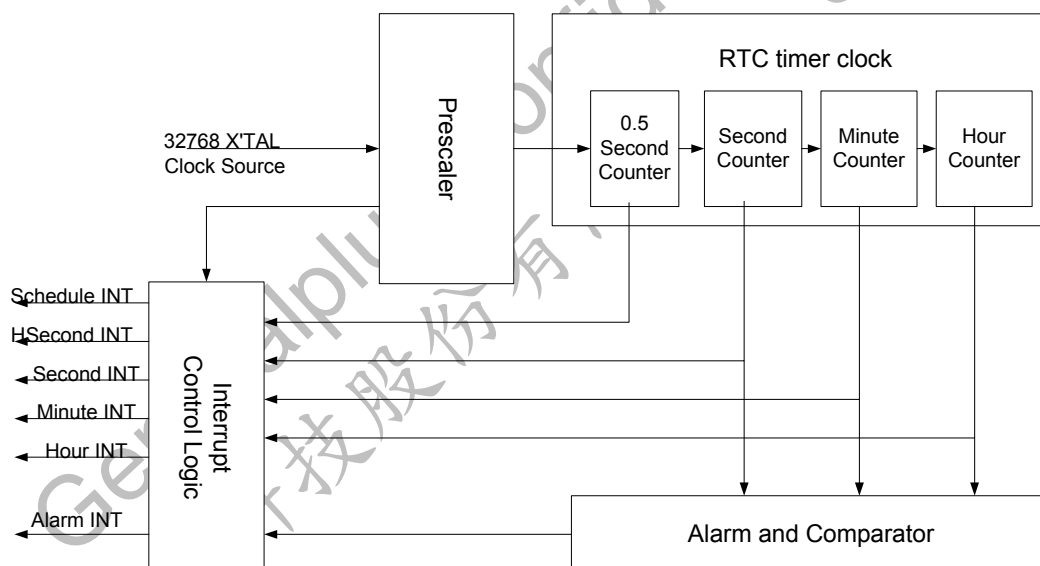
9 Real Time Clock (RTC)

9.1 Introduction

GPL162002A/162003A provides a real-time clock (RTC) module, which offers:

- Auto-update, up-to-hour clock register
- One alarm comparison registers
- Hour / Minute / Second / Half second interrupts
- Alarm Interrupt
- Scheduler interrupt: 16Hz / 32Hz / 64Hz / 128Hz / 256Hz / 512Hz / 1024Hz / 2048Hz

9.2 RTC Structure and clock source



9.3 Control Registers

Real Time Clock Control Register Summary Table

Name	Address	Description
P_Second	0x7920	Second Register
P_Minute	0x7921	Minute Register
P_Hour	0x7922	Hour Register
P_Alarm_Second	0x7924	Alarm Second Register
P_Alarm_Minute	0x7925	Alarm Minute Register
P_Alarm_Hour	0x7926	Alarm Hour Register
P_RTC_Ctrl	0x7934	HMS / Alarm / Scheduler Control Register

Name	Address	Description
P_RTC_INT_Status	0x7935	HMS / Alarm / Scheduler Interrupt Flag & Clear Register
P_RTC_INT_Ctrl	0x7936	HMS / Alarm / Scheduler Interrupt Control Register
P_RTC_HMSBusy	0x7937	RTC HMS controller busy register

P_Second
0x7920
Second Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCSEC					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	RTCSEC	R/W	Real Time Clock Second Setup Register	Only 0x00~0x3B Valid (0~59)

P_Minute
0x7921
Minute Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCMIN					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	RTCMIN	R/W	Real Time Clock Minute Setup Register	Only 0x00~0x3B Valid (0~59)

P_Hour
0x7922
Hour Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	RTCHR				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
[4:0]	RTCCHR	R/W	Real Time Clock Hour Setup Register	Only 0x00~0x17 Valid (0~23)

When updating above registers, programmers should check P_RTC_HMSBusy (0x7937) for keeping hour/minute/second setup stable.

P_Alarm_Second
0x7924
Alarm Second Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMSEC					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	ALMSEC	R/W	Alarm Second Setup Register	Only 0x00~0x3B Valid (0~59)

P_Alarm_Mintue 0x7925 Alarm Minute Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMMIN					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	ALMMIN	R/W	Alarm Minute Setup Register	Only 0x00~0x3B Valid (0~59)

P_Alarm_Hour 0x7926 Alarm Hour Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	ALMHR				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
[4:0]	ALMHR	R/W	Alarm Hour Setup Register	Only 0x00~0x17 Valid (0~23)

P_RTC_Ctrl 0x7934 HMS / Alarm / Schedule Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RTCEN	-	-	-	-	ALMEN	HMSSEN	SCHEN	-	-	-	-	-	-	SCHSEL	
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RTCEN	R/W	RTC Module Enable. If this bit is set to "1", this RTC module and Alarm Module will be enabled. If this bit is cleared to "0", RTC module is disabled to save power.	0= Disabled 1= Enabled
[14:11]			Reserved	
10	ALMEN	R/W	Alarm Function Enable. If this bit is set to "1", this alarm function will be enabled, and vice versa.	0= Disabled 1= Enabled
9	HMSSEN	R/W	Hour/Minute/Second Function Enable. If this bit is set to "1", this hour/minute/second automatic updating function will be enabled, and vice versa.	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
8	SCHEN	R/W	Scheduler Function Enable. If this bit is set to "1", this scheduler module will be enabled. If this bit is cleared to "0", scheduler module is disabled to save power.	0= Disabled 1= Enabled
[7:3]			Reserved	
[2:0]	SCHSEL	R/W	Schedule Time Period Selection. These 3 control bits are valid only when SCHEN control bit is set to "1".	000 = 16Hz 001 = 32Hz 010 = 64Hz 011 = 128Hz 100 = 256Hz 101 = 512Hz 110 = 1024Hz 111 = 2048Hz

P_RTC_INT_Status 0x7935 HMS/ Alarm / Schedule Interrupt Flag & Clear Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	ALMIEF/C	-	SCHIF/C	-	-	-	-	HRIF/C	MINIF/C	SECIF/C	HSECIF/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
10	ALMIF/C	R/W	Alarm Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the alarm interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
9			Reserved	
8	SCHIF/C	R/W	Schedule Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the scheduler interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
[7:4]			Reserved	
3	HRIF/C	R/W	Hour Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the hour interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
2	MINIF/C	R/W	Minute Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the minute interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
1	SECIF/C	R/W	Second Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect

Bit	Function	Type	Description	Condition
			second interrupt is asserted.	Write 1= Clear the flag
0	HSECIF/C	R/W	Half Second Interrupt Flag / Clear. Write "1" to clear the flag. This bit is set to "1" by hardware if the half-second interrupt is asserted.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag

P_RTC_INT_Ctrl 0x7936 HMS / Alarm / Schedule Interrupt Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	ALMIEN	-	SCHIEEN	-	-	-	-	HRIEN	MINIEN	SECIEN	HSECIEN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
10	ALMIEN	R/W	Alarm Interrupt Enable. If this bit is set to "1", and alarm interrupt occurs, hardware will issue an IRQ7 to CPU. If this bit is cleared to "0", this interrupt will be masked off.	0= Disabled 1= Enabled
9			Reserved	
8	SCHIEEN	R/W	Scheduler Interrupt Enable. If this bit is set to "1", and schedule interrupt occurs, hardware will issue an IRQ6 to CPU. If this bit is cleared to "0", this interrupt will be masked off. Generalplus suggests programmers do not use scheduler as halt/sleep mode wake-up source because the scheduler interrupt occurs more quickly than the time that CPU wakes up from halt/sleep mode. As a result, the scheduler interrupt flag will not be held from halt/sleep wake-up.	0= Disabled 1= Enabled
[7:4]			Reserved	
3	HRIEN	R/W	Hour Interrupt Enable. If this bit is set to "1" and hour interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to "0", this interrupt will be masked off.	0= Disabled 1= Enabled
2	MINIEN	R/W	Minute Interrupt Enable. If this bit is set to "1" and minute interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to "0", this interrupt will be masked off.	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
1	SECIEN	R/W	Second Interrupt Enable. If this bit is set to "1" and second interrupt occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to "0", this interrupt will be masked off.	0= Disabled 1= Enabled
0	HSECIEN	R/W	Half Second Interrupt Enable. If this bit is set to "1" and half second interrupts occurs, hardware will issue an HMS IRQ7 to CPU. If this bit is cleared to "0", this interrupt will be mask.	0= Disabled 1= Enabled

P_RTC_HMSBusy
0x7937
RTC HMS Busy Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SEC_BUSY	MIN_BUSY	HR_BUSY	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15]	SEC_BUSY	R	RTC Second Controller Busy Flag When this bit is 1, it means the RTC is busy on writing second to the register. Programmers must wait until this bit is 0 in order to write further data to second register or shut down the system clock.	0 = Not Busy 1 = Busy
[14]	MIN_BUSY	R	RTC Minute Controller Busy Flag When this bit is 1, it means the RTC is busy on writing minute to the register. Programmers must wait until this bit is 0 in order to write further data to minute register or shut down the system clock.	0 = Not Busy 1 = Busy
[13]	HR_BUSY	R	RTC Hour Controller Busy Flag When this bit is 1, it means the RTC is busy on writing hour to the register. Programmers must wait until this bit is 0 in order to write further data to hour register or shut down the system clock.	0 = Not Busy 1 = Busy
[12:0]			Reserved	

9.4 Program Examples

```

_RESET:

// Setup Current Clock as 07:59:59 (H:M:S)

secondwait?:
    r1=[P_RTC_Busy]                // After writing new value to P_Second, it must
    test r1,0x8000                // wait RTC second idle state to make sure the
    jnz secondwait?               //new value write valid.
    r1=59
    [P_Second]=r1

minutewait?:
    r1=[P_RTC_Busy]                // After writing new value to P_Minute, it must
    test r1,0x4000                // wait RTC minute idle state to make sure the //new
    jnz minutewait?               value is written valid.
    r1=59
    [P_Minute]=r1

hourwait?:
    r1=[P_RTC_Busy]                // After writing new value to P_Hour, it must
    test r1,0x2000                // wait RTC hour idle state to make sure the //new
    jnz hourwait?                 value write valid.
    r1=7
    [P_Hour]=r1

    r1 = 0                        // Setup Alarm Clock as 08:00:00 (H:M:S)
    [P_Alarm_Second] = r1
    [P_Alarm_Minute] = r1
    r1 = 8
    [P_Alarm_Hour] = r1
    r1 = 0x0400                   // Enable Alarm Interrupt only
    [P_RTC_INT_Ctrl] = r1
    r1 = 0x8700                   // Enable Alarm, Scheduler and
    [P_RTC_Ctrl] = r1             // Hour/Minute/Second/Half-Second Module
    int irq                       // Enable CPU IRQ function
    jmp $                         // Dead Loop

//*****

_IRQ7:
    push r1,r2 to [sp]
    r2 = [P_INT_Status2]          // Note that even though that scheduler and
    r1 = r2 & C_INT_Alarm         // Hour/Minute/Second/Half-Second interrupt
    jz L_End_Alarm_ISR?          // are not enabled, their interrupt flags can still
    r1 = [P_RTC_INT_Status]      // be polled ( if interrupt event occurs,

```

```
        cmp r1 , 0x050F                // corresponding flag is read as "1" )
        jne  L_RTCTestError?
        DisplayResultCode  D_OK
        jmp  L_End_Alarm_ISR?
L_RTCTestError?:
        DisplayResultCode  D_NG        // Note that DisplayResultCode is a MACRO
L_End_Alarm_ISR?:                    // and Delay_Loop is a two 32768Hz clock
        pop      r1,r2 from [sp]      // cycles.
        reti
```

10 Audio Output

10.1 DAC

GPL162002A/162003A supports the function for speech and melody synthesis. The sound data can be played back in the sequence of the control functions as designed by users' program. Several algorithms are recommended for sound compression: PCM, LOG PCM, DM and ADPCM. In addition, Generalplus provides SUBBAND, CLEP, LPC, HASC, LRC, Wavetable and FM low bit rate (high compression rate) software algorithm on this GPL162002A/162003A Integrated Development Environment (IDE).

The GPL162002A/162003A has a 16-bit stereo D/A converter with headphone amplifier circuitry which can drive 16Ω headphone directly. All voice data will be converted to data of 48K Hz sample rate by the embedded Sample Rate Controller, SRC. Programmers can use the internal EQ and 3D effector to increase sound quality. GPL162002A/162003A provides two 16-bit DAC drivers for two channel audio outputs. The audio driver can be amplified by a bipolar junction transistor or by an amplifier to drive a set of speaker or buzzer. GPL162002A/162003A also supports IIS interface for connecting other AC device.

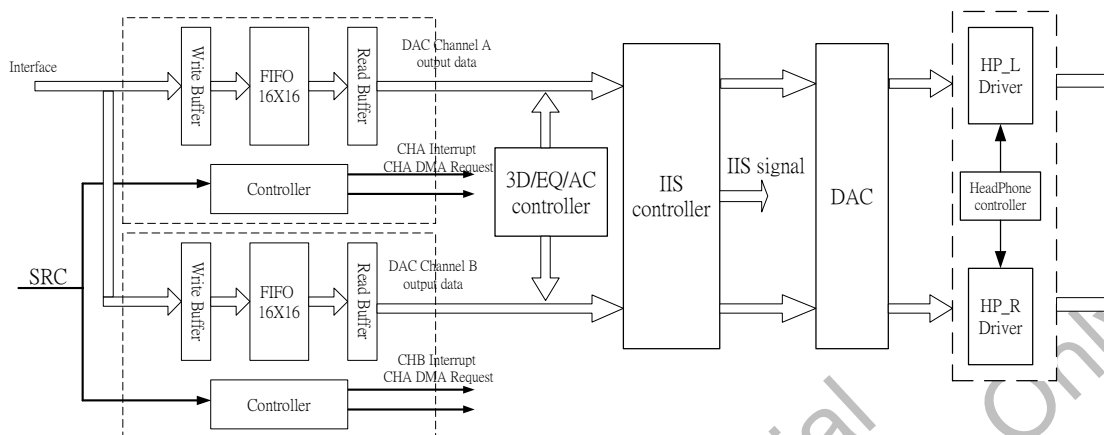
- Two 16-bit DAC channels
- Each channel has a 16 X 16-bit ring buffer (FIFO)
- FIFO empty interrupt
- FIFO full indication flag
- Support IIS mode
- Sample rate converter which can convert voice to data of 48K Hz sample rate
- Embedded digital 7-band equalizer (EQ)
- Embedded a 3D surround effector

10.2 DAC Operation

In DAC mode, DACA signal is outputted through DAC_L pin, and the DACB signal is through DAC_R pin. The data of DACA and DACB should be delivered to P_CHA_Data (0x78F1) and P_CHB_Data (0x78F9) registers, respectively.

In the DAC mode, external components (some amplification circuit) are necessary to drive a speaker. There are several solutions for these external components, for example, simple transistors (8050BJT) or standard OPs (LM324) or GENERALPLUS amplifier GPY0030.

10.3 Block Diagram



10.4 Speech Mode

In GPL162002A/162003A, data written to data control register, P_CHA_Data or P_CHB_Data, will be saved in FIFO buffer. To playback voice or audio sound, programmers need to obtain appropriate PCM data and then write them to corresponding data control register with certain frequency, known as sample rate, such as 8 KHz. When the corresponding timer overflows, the audio controller will send the data in FIFO to audio output. In GPL162002A/162003A, SRC controls the sample rate for CHA and CHB, respectively. To obtain appropriate PCM data depends on the decompression algorithm programmer chosen.

GPL162002A/162003A has a built-in 7-band EQ. These 7-band center frequencies are {100, 150, 400, 1K, 3K, 7K, and 15K}. Programmers can define music style by themselves through setting this 7-band EQ. Generalplus also provides some EQ designated value for reference. Please refer to the following descriptions for details.

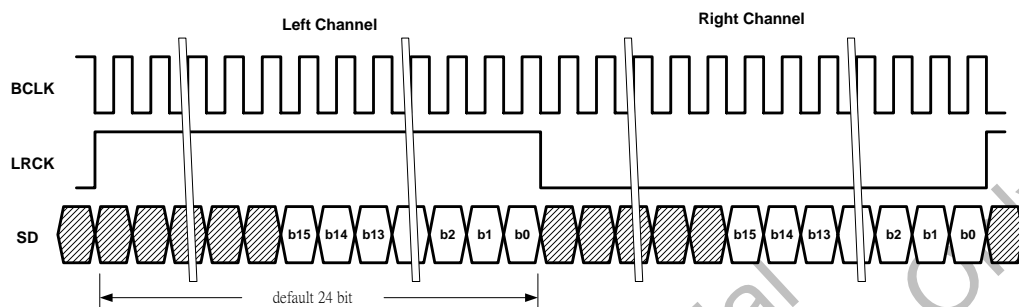
GPL162002A/162003A also has a built-in 3D surround effector to increase sound stereo quality. It is easy to set up the 3D effector in GPL162002A/162003A by just configuring 5 corresponding control registers.

10.5 IIS Mode

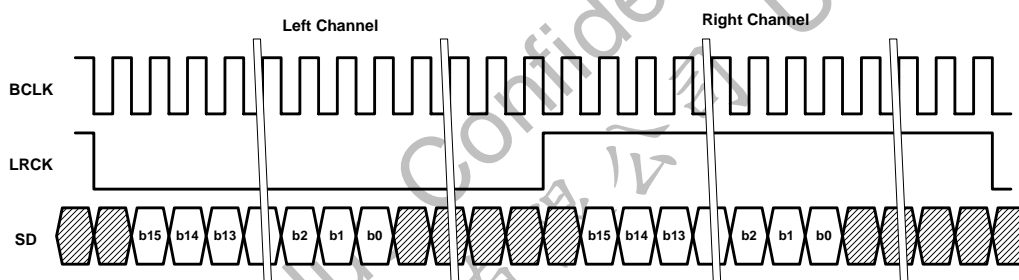
GPL162002A/162003A supports IIS interface of 4 physical signal lines to connect other DAC devices.

- BCLK, bit clock output, shared with IOC14
- LRCK, left/ right select output, shared with IOC13
- DA, data output, shared with IOC12
- MCLK, main clock, share with IOC15

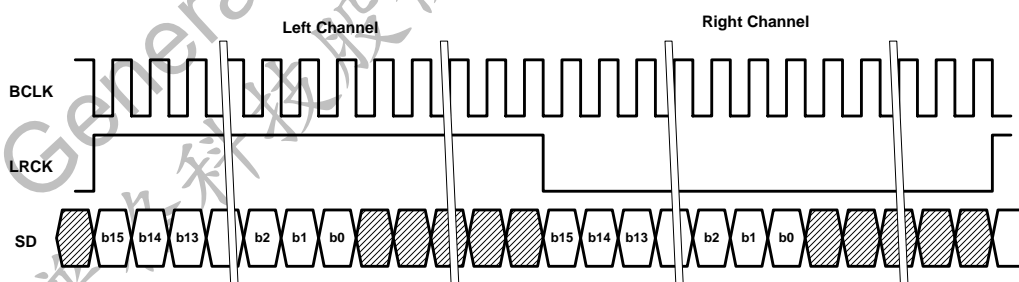
There are three types of IIS transmission format on GPL162002A/162003A. These transmission formats are selected by given designated value to P_DAC_IIS_Mode (0x78FF. b [3:2]) register. The details of setting transmission modes and timing diagram are as below.



a. Right justified mode



b. shift 1 bit mode



c. left justified mode

10.6 Control Registers

DAC Control Register Summary Table

Name	Address	Description
P_CHA_Ctrl	0x78F0	Channel A Control Register
P_CHA_Data	0x78F1	Channel A Data Register
P_CHA_FIFO	0x78F2	Channel A FIFO Control Register
P_CHB_Ctrl	0x78F8	Channel B Control Register
P_CHB_Data	0x78F9	Channel B Data Register
P_CHB_FIFO	0x78FA	Channel B FIFO Control Register
P_DAC_Ctrl	0x78FD	DAC Control Register
P_HPAMP_Ctrl	0x78FE	Headphone Amplifier Control Register
P_DAC_IIS_Ctrl	0x78FF	DAC IIS Mode Control Register
P_DAC_ACCREQ	0x7BF0	3D/EQ/AC Parameter Access Request Register
P_DAC_ACCDINL	0x7BF1	3D/EQ/AC Parameter Data Input Low Register
P_DAC_ACCDINH	0x7BF2	3D/EQ/AC Parameter Data Input High Register
P_DAC_EFF_Ctrl	0x7BF3	3D/EQ/AC Control register
P_DAC_ACTHRESL	0x7BF4	AC Threshold Low register
P_DAC_ACTHRESH	0x7BF5	AC Threshold High register
P_DAC_EQBANDSEL	0x7BF6	EQ Band Index Selection
P_DAC_EQSPEC	0x7BF7	EQ Band Spectrum output
P_DAC_VOLUME3D	0x7BF8	3D Main Volume
P_DAC_VOLUME3D_C	0x7BF9	3D Center Volume
P_DAC_VOLUME3D_S	0x7BFA	3D Surround Volume
P_DAC_VOLUME3D_R	0x7BFB	3D Right Channel Volume
P_DAC_VOLUME3D_L	0x7BFC	3D Left Channel Volume
P_DAC_ACCDOUTL	0x7BFE	3D/EQ/AC Parameter Data Output Low Register
P_DAC_ACCDOUTH	0x7BFF	3D/EQ/AC Parameter Data Output High Register

P_CHA_Ctrl		0x78F0						CHA DAC/PWM Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FEMIF/C	FEMIEN	CHAEN	-	SIGNED	SRCEN	SRCRST	-	-	-	-	-	SRCFS			
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FEMIF/C	R/W	FIFO Empty Interrupt Flag. Write "1" to clear the flag. This bit is set to "1" by hardware if the FIFO empty interrupt is asserted. CHAFEILV defines the level that FIFO is considered as "empty".	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
14	FEMIEN	R/W	FIFO Empty Interrupt enable. If this bit is set to "1" and FIFO empty interrupt occurs, hardware will issue an IRQ0 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ0 and FIQ, please refer to Chapter Interrupt .	0= Disable 1= Enable
13	CHAEN	R/W	CHA Enable	0= Disable 1= Enable
12			Reserved	
11	SIGNED	R/W	Input Data Formate Control.	0= Unsigned Data Input 1= Signed data Input
10	SRCEN	R/W	SRC Mode Enable. Programmer must enable this bit to activate DAC sample rate.	0= Disable SRC controller 1= Enable SRC controller
9	SRCRST	R/W	Reset SRC If this bit is set to 1, SRC will be reset. And it will be cleared to 0 after the SRC reset is done. Before turning on the SRC, programmers must write 1 to this bit.	0= No effect 1= Reset SRC
[8:4]			Reserved	
[3:0]	SRCFS	R/W	Input Sample Rate Setup. This register is valid only when SRCEN is set to 1.	0000: 44.1KHz 0001: 48KHz 0010: 32KHz 0011: 22.05KHz 0100: 24KHz 0101: 16KHz 0110: 11.25KHz 0111: 12KHz 1000: 8KHz 1001~1111: Reserved

P_CHA_Data
0x78F1
CHA DAC/PWM Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHADATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHADATA	R/W	Channel A Data Register	

P_CHA_FIFO		0x78F2								CHA FIFO Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		FFUL	FUDN	-	-	-	-	-	FRST	CHAFEILV				CHAFINX			
Default		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FFUL	R	CHA FIFO full flag. This flag is set to "1" by hardware if the FIFO is full. If the flag is "1", any data written via P_CHA_Data is invalid.	0= FIFO is not full 1= FIFO is full
14	FFUNRN	R	CHA FIFO under run flag. This flag is set to "1" by hardware if the FIFO is under running. If the flag is "1", the output of DAC is keep last output value.	0= FIFO is not under running 1= FIFO is under running
[13:9]			Reserved	
8	FRST	W	FIFO Reset	Write 0 = No effect Write 1 = FIFO reset
[7:4]	CHAFEILV	R/W	CHA FIFO Empty Interrupt Level. These control bits are used to set FIFO empty interrupt timing that is enabled on FEMIEN of P_CHA_Ctrl register. It defines the number of data left in FIFO to be considered as empty by hardware. The larger the value is, the higher frequency of FIFO empty interrupt occurs. The smaller the value is, the less frequency of the FIFO empty interrupt happens. Consequently, it saves the CPU bandwidth.	FIFO Empty Interrupt issue timing. 0000= Reserved 0001= when # of data in FIFO < 1 0010= when # of data in FIFO < 2 0011= when # of data in FIFO < 3 . . . 1110= when # of data in FIFO < 14 1111= when # of data in FIFO < 15
[3:0]	CHAFINX	R	CHA FIFO used (Default = 0000) FIFO is a 16X16-bit ring buffer.	0000= 0 data is in FIFO 0001= 1 data is in FIFO 0010= 2 data is in FIFO . . . 1110= 14 data is in FIFO 1111= 15 data is in FIFO

P_CHB_Ctrl		0x78F8					CHB DAC/PWM Control Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	FEMIF/C	FEMIEN	CHBEN	SSF	CHACFG	MONO	-	-	-	-	-	-	-	-	-	-	
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	FEMIF/C	R/W	FIFO Empty Interrupt Flag. Write "1" to clear the flag. This bit is set to "1" by hardware if the FIFO empty interrupt is asserted. CHBFEILV defines the level that FIFO is considered as "empty".	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1= Clear the flag
14	FEMIEN	R/W	FIFO Empty Interrupt enable. If this bit is set to "1", and if FIFO empty interrupt occurs, hardware will issue an IRQ0 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ0 and FIQ, please refer to Chapter Interrupt .	0= Disabled 1= Enabled
13	CHBEN	R/W	CHB Enable	0= Disabled 1= Enabled
12	SSF	R/W	CHB service frequency. If users enable channel B, this bit must be set to 1. And then the CHB will have the same sample rate with CHA.	0= Not used. 1= The same service frequency with CHA
11	CHACFG	R/W	CHB uses CHA's configuration. When CHA and CHB have the same service frequency (SSF=1), if CHACFG is set to high, the CHB will use CHA's configuration, such as CHAEN, CHA_DATA, CHA_FEILV, CHA_FRST, and CHA DMA request, so CHA and CHB can share the same DMA channel.	0=Use CHB's configuration 1=Use CHA's configuration
10	MONO	W	Monochrome mode. When CHA and CHB have the same service frequency (SSF=1) and configuration (CHACFG=1), and MONO is set to stereo mode, then write a sequence of data to CHA FIFO. As a result, the odd data will be sent to channelA and the even data will be sent to channelB over and over automatically by hardware.	0= Stereo 1= Monochrome

Bit	Function	Type	Description	Condition
			If MONO is set to Monochrome mode, the written voice data head to CHA FIFO and CHB FIFO respectively.	
[9:0]			Reserved	

P_CHB_Data
0x78F9
CHB DAC/PWM Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHBDATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHBDATA	R/W	Channel B Data Register	

P_CHB_FIFO
0x78FA
CHB FIFO Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FFUL	FUDN	-	-	-	-	-	FRST	CHBFEILV				CHBFINX			
Default	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FFUL	R	CHB FIFO full flag. This flag is set to "1" by hardware if the FIFO is full. If the flag is "1", any data written via P_CHB_Data is invalid.	0= FIFO is not full 1= FIFO is full
14	FFUNRN	R	CHB FIFO under run flag. This flag is set to "1" by hardware if the FIFO is under running. If the flag is "1", the output of DAC is keep last output value.	0= FIFO is not under running 1= FIFO is under running
[13:9]			Reserved	
8	FRST	W	FIFO reset	Write 0 = No effect Write 1 = FIFO reset.
[7:4]	CHBFEILV	R/W	CHB FIFO Empty Interrupt Level. These control bits are used to set FIFO empty interrupt timing that is enabled on FEMIEN of P_CHB_Ctrl register. It defines the number of data left in FIFO to be considered as empty by hardware. The larger the value is, the higher frequency of the FIFO empty interrupt happens. The	FIFO Empty Interrupt issue timing 0000= Reserved 0001= when # of data in FIFO < 1 0010= when # of data in FIFO < 2 0011= when # of data in FIFO < 3 . . . 1110 when # of data in FIFO < 14

Bit	Function	Type	Description	Condition
			smaller the value is, the less frequency of the FIFO empty interrupt occurs. Consequently, it saves the CPU bandwidth.	1111 when # of data in FIFO < 15
[3:0]	CHBFINX	R	CHB FIFO used (Default = 0000) FIFO is a 16X16-bit ring buffer.	0000= 0 data is in FIFO 0001= 1 data is in FIFO 0010= 2 data is in FIFO 0011= 3 data is in FIFO . . . 1110= 14 data is in FIFO 1111= 15 data is in FIFO

P_DAC_Ctrl
0x78FD
DAC Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	BPFIR	AS_S	SP_DLY	AS_CYCLE	AS_RANGE	PWDAL	PWDAR	IIS	DACLK			
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	BPFIR	R/W	Bypass DAC Digital Filter. This control bit is to set whether audio data bypass the internal digital filter.	0= Not bypass 1= Bypass
10	AS_S	R/W	Auto Sleep Function Input Source Select. This control bit is to select input sources of auto sleep function which is either from IIS channel directly or from IIS channel then through digital filter.	0: IIS output 1: Digital filter output
[9:8]	SP_DLY	R/W	Time Delay Between DAC and Heaphphone. These control bits are to set delay time between DAC and headphone driver when DAC is ON/OFF. This is to avoid "bo" sound when turning ON/OFF DAC function. For ON procedure, turn on DAC first then headphone driver circuitry; for OFF procedure, turn off DAC after headphone driver circuitry.	00: 1440 * DALRC 01: 2880 * DALRC 10: 4320 * DALRC 11: 5760 * DALRC Note: DALRC is 48KHz
[7:6]	AS_CYCLE	R/W	Auto Sleep Start-up Time. When auto sleep condition exists, hardware will turn off DAC for designated sleep time to reduce white noise.	00: 8192 * DALRC 01: 16384 * DALRC 10: 32768 * DACRC 11: 65536 * DALRC

Bit	Function	Type	Description	Condition
				Note: DALRC is 48KHz
[5:4]	AS_RANGE	R/W	Auto Sleep code Variation Range. These bits are to set auto sleep code level condition. When auto sleep function input data matches AS_RANGE setup value, auto sleep condition exists.	00: 0 01: -1~0 10: -1~1 11: -2~2
3	PWDAL	R/W	DAC Left Channel Power Control.	0: Operation mode 1: Power-down mode
2	PWDAR	R/W	DAC Right Channel Power Control.	0: Operation mode 1: Power-down mode
1	IIS	R/W	Internal IIS Format Select. This control bit is to set internal IIS mode. The IIS mode setting must be the same with setting of IIS_MODE (0x78FF.b [3:2]). Besides, if using internal DAC IIS mode, the IIS left justified mode (0x78FF.b [3:2] =10) is not supported.	0: right justified mode 1: shift 1 bit mode
0	DACLK	R/W	DAC Main Clock Enable. The DAC needs an 18.432MHz clock for digital filter. Before enabling the DAC. Programmers must enable the DAPLL (0x7807.b4) and wait it stable and then set this bit to 1 to initialize DAC clock.	0: Disable DAC clock 1: Enable DAC clock

P_HPAMP_Ctrl 0x78FE Headphone Amplifier Control Register

PWSPVR												SPINS					PWSPL				PWSPR	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
Function	-	-	-	-	-	-	-	-	-	-	-	PWSPVR	SPINS	SPINS	PWSPL	PWSPR						
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1						

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
4	PWSPVR	R/W	Headphone Direct Driver Control. This bit set to 0 will enable headphone power driver. The hardware can drive a set of headphone directly but needs external audio amplifier circuitry to drive a set of speaker.	0= Headphone power on 1= Power-down mode
[3:2]	SPINS	R/W	Audio Driver Input source select. If SPINS=00, then audio driver output data are from DAC. If SPINS=01, audio driver is internally connected to ADC line-in channel so that hardware can output	00: DAC output 01: Line-in volume output 10: MIC output 11: Reserved

Bit	Function	Type	Description	Condition
			recorded data directly from line-in channel. If SPINS=10, audio driver is internally connected to ADC MIC channel so that hardware can output recorded data directly from MIC channel.	
1	PWSPL	R/W	Headphone Left Channel Power Control Register. This bit available only when SPINS=11b. If SPINS=00, the left channel power controlled by P_DAC_Ctrl b3.	0= Left channel power on 1= Power-down
0	PWSPR	R/W	Headphone Right Channel Power Control Register. This bit available only when SPINS=11b. If SPINS=00, the right channel power controlled by P_DAC_Ctrl b2.	0= Right channel power on 1= Power-down

P_DAC_IIS_Ctrl
0x78FF
DAC IIS Mode Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	IIS_MCLK	IIS_BITS	IIS_MODE	IISEXT	IISEN			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:6]	IIS_MCLK	R/W	IIS External DAC Main Clock Selection. These control bits are to set IIS_MCLK clock which is a must for a high-level DAC. These bits are valid only when IISEN and IISEXT are both set to 1.	00: 384 * FS 01: 256 * FS 10: 192 * FS 11: 128 * FS Note: FS is IIS_LRCK clock.
[5:4]	IIS_BITS	R/W	IIS Data Bits Control Register. If internal DAC is selected, the IIS_BITS needs to be set to 00.	00: 24 bits (internal DAC support only) 01: 16 bits 10: 32 bits 11: Reserved
[3:2]	IIS_MODE	R/W	IIS Output Mode Select. If internal DAC is selected, IIS output mode should be the same with P_DAC_Ctrl (0x78FD. b1) IIS mode.	00: Right justified 01: shift 1 bits 10: Left justified 11: Reserved
1	IISEXT	R/W	External DAC Mode Enable. This bit is valid only when IISEN is set to 1.	0: Disable 1: Enable

Bit	Function	Type	Description	Condition
0	IISEN	R/W	DAC IIS Mode Enable. This bit must be set to 1 when the internal DAC is used.	0: Disable 1: Enable

The Following registers are used to download/upload parameter to 3D or EQ/AC modules. Before enabling these modules, the correct registers must be downloaded with corresponding parameters into these modules.

P_DAC_ACCREQ 0x7BF0 3D/EQ/AC Parameter AccessRequest Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RDY	WRITE	-	-	-	-	3D	ADDR								
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RDY	R/W	Access Ready Register. Any request must be issued when RDY is 1. When RDY is 0, all requests will be discarded.	0: Not ready 1: Ready for read/write parameter
14	WRITE	R/W	Read or Write Access Register.	0: Read parameter 1: Write parameter
[13:10]			Reserved	
9	3D	R/W	3D or EQ/AC Selection.	0: Download EQ/AC parameter 1: Download 3D parameter
[8:0]	ADDR	R/W	Parameter Download Address.	

P_DAC_ACCDINL 0x7BF1 3D/EQ/AC Parameter Data Input Low Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DATAIN[15:0]															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DATAIN	R/W	Programmers must fill a designated parameter in this register before writing to P_DAC_ACCREQ register.	

P_DAC_ACCDINH 0x7BF2 3D/EQ/AC Parameter Data Input High Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DATAIN[23:16]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	DATAIN	R/W	Programmers must fill a designated parameter in this register before writing to P_DAC_ACCREQ register. In 3D mode, programmers do not need to fill any value in this register since the parameter is 16-bit.	

P_DAC_EFF_Ctrl
0x7BF3
3D/EQ/AC Control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	3DEN	2CH	HP	EQEN	BPEQ	BPAC	-	DEPTH_3D								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	3DEN	R/W	3D Module Enable Control. Before enabling 3D module, 2CH/HP/DEPTH_3D and 3D filter parameter must be set correctly.	0: Disable 1: Enable
14	2CH	R/W	2CH/5CH 3D selection.	0: 5 channel mode 1: 2 channel mode
13	HP	R/W	Headphone Output Selection.	0: Speaker output 1: Headphone output
12	EQEN	R/W	EQ/AC Module Enable Control. Before enabling EQ/AC module, the EQ/AC parameter must be set correctly.	0: Disable 1: Enable
11	BPEQ	R/W	Bypass EQ module Control.	0: Not bypass 1: Bypass
10	BPAC	R/W	Bypass AC module Control.	0: Not bypass 1: Bypass
9			Reserved	
[8:0]	DEPTH_3D	R/W	3D Surround Depth Control.	

P_DAC_ACTHRESL
0x7BF4
AC(anti-clip) Threshold Low Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ACTHRES[15:0]															
Default	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ACTHRES	R/W	AC Threshold Register [15:0]	

P_DAC_ACTHRESH		0x7BF5								AC(anti-clip) Threshold High Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	ACTHRES[23:16]							
Default		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	ACTHRES	R/W	AC Threshold Register [23:16]	

P_DAC_EQBANDSEL		0x7BF6		EQ Band Index Selection												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	EQBAND		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
[2:0]	EQBAND	R/W	EQ BandSpectrum Output Selection.	000: Select band 0 001: Select band 1 010: Select band 2 011: Select band 3 100: Select band 4 101: Select band 5 110: Select band 6 111: Resreved

P_DAC_EQSPEC					0x7BF7		EQ Band Spectrum output									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	EQSPEC											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	EQSPEC	R	EQ Spectrum of Each Band. This register represents the spectrum of band set in EQBAND.	

P_DAC_VOLUME3D		0x7BF8						3D Main Volume									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		VOL_3D															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	VOL_3D	R/W	The volume control register of all 3D module.	0000h: Mute FFFFh: maximum

P_DAC_VOLUME3D_C 0x7BF9
3D Center Volume

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VOL_3D_C															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	VOL_3D_C	R/W	The volume control register of center channel in 3D module.	0000h: Mute FFFFh: maximum

P_DAC_VOLUME3D_S 0x7BFA
3D Surround Volume

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VOL_3D_S															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	VOL_3D_S	R/W	The volume controls register of surround channel in 3D module.	0000h: Mute FFFFh: maximum

P_DAC_VOLUME3D_R 0x7BFB
3D Right Channel Volume

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VOL_3D_R															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	VOL_3D_R	R/W	The volume controls register of right channel in 3D module.	0000h: Mute FFFFh: maximum

P_DAC_VOLUME3D_L 0x7BFC
3D Left Channel Volume

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VOL_3D_L															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	VOL_3D_L	R/W	The volume control register of left channel in 3D module.	0000h: Mute FFFFh: maximum

P_DAC_ACCDOCTL 0x7BFE 3D/EQ/AC Parameter Data Output Low Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DATAOUT[15:0]															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DATAOUT	R	This is to read the 3D or EQ/AC parameter. This data will be valid after RDY of 0x7BF0 is 1.	

P_DAC_ACCDOUTH 0x7BFF 3D/EQ/AC Parameter Data Output High Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	DATAOUT	R	This data will be valid after RDY of 0x7BF0 becomes 1. In 3D mode, programmers do not need to fill any value in this register, since the parameter is 16-bit.	

10.7 Mute level

In DAC mode, the mute level is 0x0000. In order to make sound complete silence, Generalplus recommends programmers should make both channelA and channelB audio output value to mute level. In addition, to completely shut down audio output module on GPL162002A/162003A (very-low power consumption), both channelA and channelB enable control bits (P_CHA_Ctrl.bit13 and P_CHB_Ctrl.bit13) should be clear to "0" and DAC power and headphone power should be set to power down mode.

10.8 Ramp Up and Ramp Down

As we know, middle value of speech file is 0x8000 (first PCM data of speech file usually starts with middle level, 0x8000). The mute level on DAC mode is 0x0000. Therefore, in order to avoid unexpected sound caused by changing audio output value rapidly (from 0x0000 to 0x8000), some smooth methods, as known as ramp up and ramp down, should be involved.

Ramp up is the operation that changes from mute level to middle level (0x8000) while Ramp down is the operation that changes from middle level (0x8000) to mute level. The implementation of ramp up and ramp down is to achieve the target value by gradually adding/subtracting with a constant slope from the start value.

For GPL162002A/162003A, the ramp up and ramp down procedure is done by hardware. For power on ramp up procedure, programmers should turn on DAC power (0x78FD.b [3:2]) control first and wait it stable and then turn on headphone power (0x78FE.b [1:0]). For power down ramp down procedure, programmers should turn off headphone power (0x78FE.b [1:0]) and wait it power down and then turn off DAC power (0x78FD.b [3:2]).

10.9 3D/EQ/AC parameter reference setup

Generalplus provides some 3D and EQ/AC parameters for reference. Please refer to the tables below for details.

10.9.1 EQ/AC Control Method

EQ/AC Filter Coefficient Register Mapping:

Address	Suggested Value
0x20	0x3F2AEC
0x21	0x80DB59
0x22	0x3DCDD8
0x23	0x825E4A
0x24	0x3AA52B
0x25	0x86676F
0x26	0x311BE2
0x27	0x978040
0x28	0x2150F8
0x29	0xC4C1F1
0x2A	0x086CFE
0x2B	0x1BB754
0x2C	0xC0D513

Filter Gain Register Mapping:

Address	Meaning
0x40	Gain of Band 0 = G0
0x41	Gain of Band 1 = G1
0x42	Gain of Band 2 = G2
0x43	Gain of Band 3 = G3

Address	Meaning
0x44	Gain of Band 4 = G4
0x45	Gain of Band 5 = G5
0x46	Gain of Band 6 = G6

By adjusting G0~G6, many kinds of effects can be applied to output voice. There are 25 setps for setting gain. The following table shows the setting value in each step.

Step	Value	Step	Value	Step	Value
0	0x0809BC	10	0x196B23	20	0x50615F
1	0x0904D1	11	0x1C8520	21	0x5A3031
2	0x0A1E89	12	0x200000	22	0x653161
3	0x0B5AA1	13	0x23E793	23	0x718A50
4	0x0CBD4B	14	0x28491E	24	0x7F64F0
5	0x0E4B3B	15	0x2D3382		
6	0x1009B9	16	0x32B772		
7	0x11FEB3	17	0x38E7AA		
8	0x1430CD	18	0x3FD930		
9	0x16A77D	19	0x47A39A		

The follow effects can be applied by setting different gain in each band.

Effect	{G0, G1, G2, G3, g4, G5, G6}
DBB	{18, 18, 16, 16, 12, 9, 5}
ROCK	{17, 17, 15, 8, 2, 8, 18}
JAZZ	{18, 16, 14, 1, 16, 18, 17}
POP	{12, 18, 4, 12, 18, 12, 8}
LIVE	{7, 10, 14, 17, 18, 10, 7}

10.9.2 3D Control Method

Filter Coefficient Register Mapping:

Address	Definition
0x000~0x0FF	HRTF Filter Coefficient
0x100~0x1FF	CSS Filter Coefficient

10.10 Program Examples

```

r1 = 0x8410                                // Setup PLL Frequency as 48MHz,
[P_Clock_Ctrl] = r1                        //enable AD/DA clock source

r1=0x8200

```

```

[P_CHA_Ctrl]=r1                // Clear CHA FIFO interrupt flag, reset
r1=0x0120                      //SRC controller
[P_CHA_FIFO]=r1                // Set CHA FIFO empty interrupt level

r1=0x9000
[P_CHB_Ctrl]=r1                // Clear CHB FIFO interrupt flag, set the
r1=0x0120                      //same sample rate with CHA
[P_CHB_FIFO]=r1                // Set CHB as Stereo mode
                                // Set CHB FIFO empty interrupt level

r1=[ P_CHA_Ctrl]
r1|=0x03                      // Setup sample rate Overflow Frequency
[P_CHA_Ctrl]=r1                // as 22.05KHz

R1=0x01
[P_DAC_Ctrl]=r1                //enable DAC clock, R-L channel power
                                //on

R1=0x00                        //enable headphone R-L power
[P_HPAMP_Ctrl]=r1

R1=0x01                        //IIS enable
[P_DAC_IIS_Ctrl]=r1

r1=[ P_CHA_Ctrl]
r1|=0x6400                    //enable CHA, enable CHA interrupt
[P_CHA_Ctrl]=r1

r1=[ P_CHB_Ctrl]
r1|=0x2000                    //enable CHB
[P_CHB_Ctrl]=r1

irq on
.....

//*****
_IRQ0:
.....
push r1 to [sp]
r1=[P_INT_Status1]
test  r1,INT_AudioA
jz    CHB?

CHA?:

r1=[P_CHA_Ctrl]
[P_CHA_Ctrl]=r1
ReadResourceData_A    D_Sound_A    // Clear Channel A FIFO Empty Interrupt
[P_CHA_Data]=r1

```

```
CHB?:                                     // Fetch CHA Data

ReadResourceData_B    D_Sound_B
[P_CHB_Data]=r1

                                     // Fetch CHB Data

jmp endirq0?

pop r1 from [sp]

endirq0?:                                reti

                                     // Note: ReadResourceData is MACRO
```


11 STN LCD

11.1 Introduction

The GPL162002A/162003A contains a powerful STN LCD controller and it can support resolution up to 320(H) X 320(V) and support 16 gray levels for monochrome STN or 4096 colors for color STN. The LCD controller also has a built-in hardware scroll function to reduce software overhead. Moreover, the interface supports flexible 1-bit, 4-bit or 8-bit data interface to connect with a variety of LCD panels.

The LCD controller in GPL162002A/162003A has the following features:

- Supports standard STN LCD panel driver interface.
- Built-in frame rate control for gray and color display.
- Supports standard 1/ 4/ 8-bit LCD driver Interface.
- Supports monochrome, 4 gray levels, and 16 gray levels.
- Supports 2, 4, 16, 256, and 4096 colors display.
- Supports virtual display screen up to 1024 x 1024 for hardware horizontal and vertical scrolling capability.

11.2 LCD Control Pin Configuration

Name	I/O	Description
LCDFM	O	Frame modulation signal (shared with GPIO PortC0)
LCDFP	O	Frame Rate signal (shared with GPIO PortC1)
LCDLP	O	Line Scan signal (shared with GPIO PortC2)
LCDCP	O	Shifting clock signal (shared with GPIO PortC3)
LCDD0	O	LCD Data 0 (shared with PortA0)
LCDD1	O	LCD Data 1 (shared with PortA1)
LCDD2	O	LCD Data 2 (shared with PortA2)
LCDD3	O	LCD Data 3 (shared with PortA3)
LCDD4	O	LCD Data 4 (shared with PortA4)
LCDD5	O	LCD Data 5 (shared with PortA5)
LCDD6	O	LCD Data 6 (shared with PortA6)
LCDD7	O	LCD Data 7 (shared with PortA7)

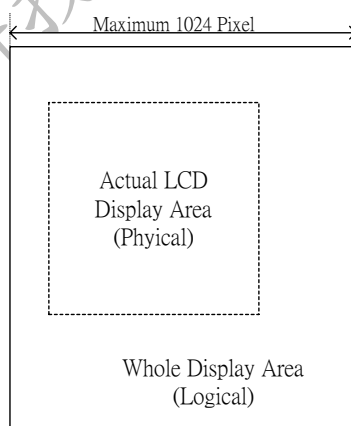
The LCDFM, LCDFP, LCDLP, and LCDCP are synchronous signals used to transfer LCD display data from GPL162002A/162003A to GPLDs or SPLCs. The LCDD [7:0] is a data port and responsible for carrying transferred LCD data.

11.3 LCD Buffer

GPL162002A/162003A allows programmers to locate a LCD buffer inside entire 64M-word addressing range of u'nSP CPU. In other words, programmers are able to define any area within 64M-word addressing field as a LCD buffer. Certainly, the area, defined as a LCD buffer, can be SRAM, ROM, or Flash or can also be internal RAM. The control registers, P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr, define the start address of LCD buffer. In addition, different LCD pages (LCD buffer) can be easily switched by programming the LCD start address register. Programmers can define more than one LCD buffer in the system. In this case, the switch operation between different LCD pages can accelerate frames updating.

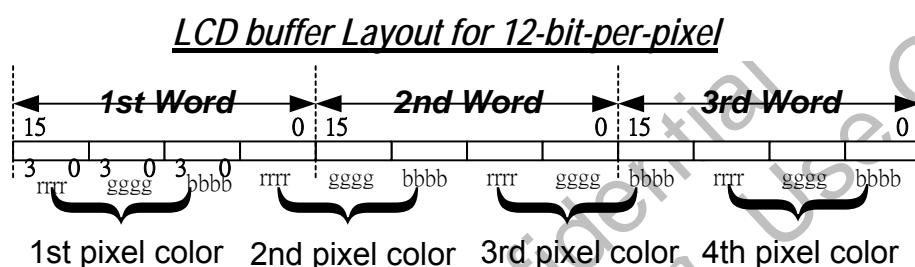
The definition of LCD buffer size is implied in two control registers, P_LCD_Common and P_LCD_Segment. The maximum number of common lines (horizontal line) to be supported is 320 pixels and the maximum number of segment lines (vertical line) is also 320 pixels as well. In most cases, the size of LCD buffer equals to $\{(P_LCD_Common + 1) \times (P_LCD_Segment + 1) \times bpp\}$ bits, where bpp represents bit-per-pixel.

GPL162002A/162003A also supports virtual page function that allows programmers to define a virtual LCD buffer larger than actual viewable size (actual LCD size). In this case, programmers can easily scroll actual LCD display screen horizontally and vertically on the full virtual page by simply defining P_LCD_Buffer_Offset control register and modifying the following two control registers, P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr. The maximum width in a virtual page is 1024 pixels, as being indicated in the following diagram.



11.4 LCD Palette

Any color on a pixel can be combined from three elements: red, green and blue. Each element, which is connected to a single LCD segment line, can be divided into 16 levels. Maximum level of display color will be 16 (red) x 16 (green) x 16 (blue), or says 4096. Therefore, one pixel takes 12 bits LCD buffer to represent a color level. We call this configuration as 12-bit-per-pixel. LCD buffer layout for each pixel is depicted as following diagram. Please note that 3-word contents 4 pixel color information.

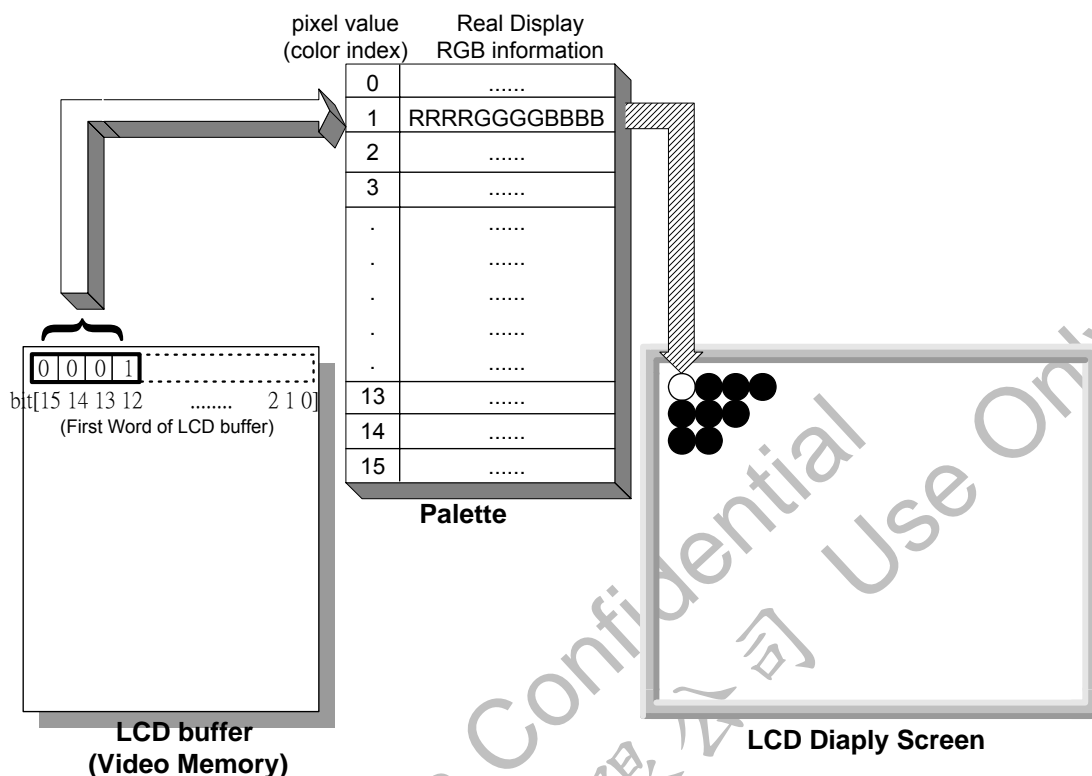


The larger the LCD resolution is (number of common X number of segment), the larger the size of LCD buffer is. In most cases, 12-bit-per-pixel configuration is not popular because it requires larger size of a LCD buffer and it is too costly. As a result, the concept of the palette is derived from these drawbacks mentioned above.

Take 4 bit-per-pixel configuration as an example: in LCD buffer, each pixel takes 4 bits, that is, each pixel can display 16 colors. However GPL162002A/162003A can display 4096 colors, now the problem becomes how we can map a 4-bit pixel value to a real display value. The solution is "Palette", the translation table.

If the palette is enabled, 4-bit value, range form 0 to 15, does not represent the real display information. Instead, this 4-bit value represents an index to the palette (mapping table). Palette is a table with the depth equal to the number of index. For example, palette depth of 4-bit-per-pixel configuration is 16; palette depth of 8-bit-per-pixel configuration is 256. Table value for each index is the real display color information (12-bit; range form 0 ~ 4095). According the mapping information in the palette, GPL162002A/162003A can automatically and easily convert the index value in LCD buffer to the real display color. Following diagram depicts the relationship between LCD buffer, palette and a real display image.

Note that if palette is activated, maximum number of display colors is the depth of Palette at one time. Certainly, programmers can change display color level in color index (pixel value) dynamically by modifying corresponding Palette registers.



GPL162002A/162003A Palette location starts with 0x7A00 of CPU view. For color configuration, the valid bit in each word of Palette is 12 bits. On the other hand, for gray level configuration, the valid bit in each word of Palette is only 4 bits. Please refer to the following table.

Control Register	Gray Level Mode								Color Mode							
	Word								word							
Location	15	12	11	8	7	4	3	0	15	12	11	8	7	4	3	0
0x7A00								$g_3g_2g_1g_0$				$R_3R_2R_1R_0$	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$		
0x7A01								$g_3g_2g_1g_0$				$R_3R_2R_1R_0$	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$		
0x7A02								$g_3g_2g_1g_0$				$R_3R_2R_1R_0$	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$		
...								$g_3g_2g_1g_0$				$R_3R_2R_1R_0$	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$		
...								$g_3g_2g_1g_0$				$R_3R_2R_1R_0$	$G_3G_2G_1G_0$	$B_3B_2B_1B_0$		

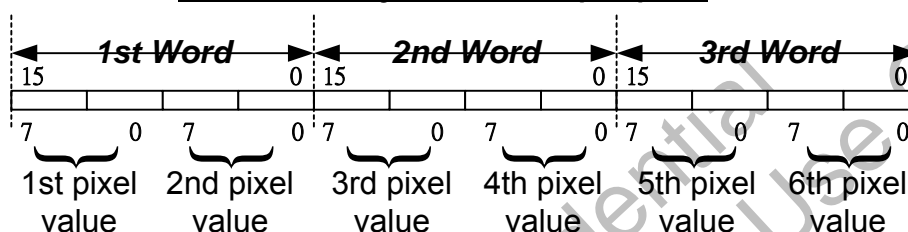
 Invalid Area (Reserved Area)

Therefore, if Palette is activated, values in a LCD buffer are only indexes of color. It does not contain the real display color information. On the contrary, if Palette is bypassed (not activated), values in a LCD buffer is the real display color information.

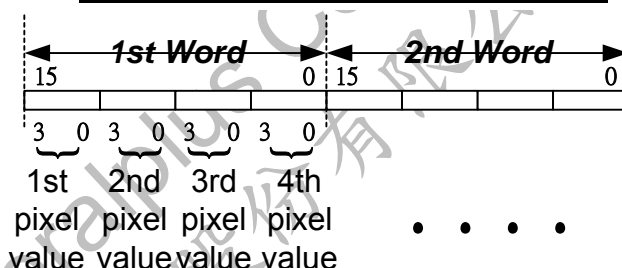
GPL162002A/162003A provides many kind of LCD configuration that varies with n-bit-per-pixel, Palette operation, and color. P_LCD_Palette_Ctrl control register will easily help programmers to set up their own configurations. Please refer to the following **Section: Control Register** for details.

The color index (with Palette) or real color level (without Palette) are sequentially distributed In a LCD buffer. The following three diagrams depict the detailed layouts for 8/4/2-bit-per-pixel.

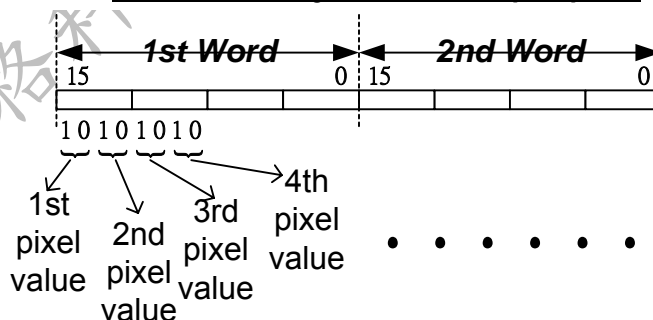
LCD buffer Layout for 8-bit-per-pixel



LCD buffer Layout for 4-bit-per-pixel



LCD buffer Layout for 2-bit-per-pixel



11.5 Control Registers

STN LCD Control Register Summary Table

Name	Address	Description
P_LCD_Setup	0x7980	LCD setup register
P_LCD_Clock	0x7981	LCD clock register

Name	Address	Description
P_LCD_Segment	0x7982	LCD segment number register
P_LCD_Common	0x7983	LCD common number register
P_LCD_Buffer_LowAdr	0x7984	LCD start address register (A0~A15)
P_LCD_Buffer_HighAdr	0x7985	LCD start address register (A16~A25)
P_LCD_Buffer_Offset	0x7986	LCD virtual page offset register
P_LCD_Timing_Ctrl	0x7987	LCD control signal timing register
P_LCD_Frame_Ctrl	0x7988	LCD frame modulation control register
P_LCD_Palette_Ctrl	0x7989	LCD Palette control register
P_LCD_Attri_Ctrl	0x798A	LCD attribute control register
P_LCD_Palette[0:255]*	0x7A00~0x7AFF	LCD 256-color palette entries

There are eleven control registers, residing from 0x7980 through 0x798A, to control the LCD Interface in GPL162002A/162003A. If color mode or gray level mode is enabled, LCD palette, locating in 0x7A00 ~ 0x7AFF, is activated. The following sections give detailed descriptions of these control registers.

* Note that Palette register(s) can be modified only when LCDEN control bit is cleared to "0". Besides, delay with one frame-rate interval is necessary before starting to modify Palette registers.

Step 1) Clear LCDEN bit to "0" // (P_LCD_Ctrl.bit13==0)
 Step 2) Delay // with one LCD-frame-rate-interval
 Step 3) Start to modify Palette register(s)
 Step 4) Set LCDEN bit to "1" // Re-enable LCDEN

P_LCD_Setup				0x7980 LCD Setup Register												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FPIF/C	FPIEN	LCDEN	SELF	-	-	BUSW		-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	FPIF/C	R/W	LCD FP Signal Interrupt Flag. This bit is set to "1" by hardware if the FP interrupt is asserted. When FP signal is at rising edge and FPIEN is set to "1", the FP interrupt is issued (FPIF=1). It informs CPU that a new frame is beginning. And CPU may update some display data to a new frame.	Read 0= Not Occurred Read 1= Occurred Write 0 = No Effect Write 1= Clear the flag
14	FPIEN	R/W	LCD FP Signal Interrupt Enable. If this bit is set to "1" and if FP interrupt is	0= Disabled 1= Enabled

Bit	Function	Type	Description	Condition
			generated, hardware will issue an IRQ5 or FIQ to CPU. If this bit is cleared to "0", the interrupt will be masked off. To select between IRQ5 and FIQ, please refer to Chapter Interrupt .	
13	LCDEN	R/W	LCD Interface Enable. If this bit is set to "1", LCD interface is enabled. Or, LCD interface is disabled. Setting LCDEN control bit to "1" will enable LCD controller. This control bit should be remained as "1" whenever the LCD interface is active. When this control bit is cleared to "0", all output signals will stay in LOW state.	0= Disabled 1= Enabled
12	SELF	R/W	Self Refresh mode. If the external LCD driver(s) involves the built-in memory and supports the self-refresh mode, the LCD interface can be configured to self-refresh mode. The LCD driver shows the last display data and GPL162002A/162003A outputs FM, FP, LP signals only and the CP and LCD Data signal will remain at ground state.	0= Disabled 1= Enabled
[11:10]			Reserved	
[9:8]	BUSW	R/W	LCD hardware data bus width configuration. 1/4/8 bit-width is supported in GPL162002A/162003A.	00= 1-bit (LCDD0 Valid) 01= 4-bit (LCDD[3:0] Valid) 10= 8-bit (LCDD[7:0] Valid) 11= Reserved
[7:0]			Reserved	

Bus Width configuration

BUSW [1:0]

00: 1 bit (LD0)

01: 4 bit (LD3~0)

10: 8 bit (LD7~0)

11: Reserved

The "V" represents support and "X" means "not support".

	BPP=1			BPP=2			BPP=4			BPP=8			BPP=12		
BUSW[1:0]	1	4	8	1	4	8	1	4	8	1	4	8	1	4	8
LCDBW=1 (B/W)	V	V	V	V	V	V	V	V	V	X	X	X	X	X	X
LCDBW=0 (Color)	X	V	V	X	V	V	X	V	V	X	V	V	X	V	V

P_LCD_Clock		0x7981						LCD Clock Generation Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	LCDCLK									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LCDCLK	R/W	Pixel Clock Divider	Range= 0 ~ 1023

$LCDCLK [9:0] = SYSCLK / (LCDCOM+1) / (LCDSEG+1) / \text{Frame-rate} - 2$

where

LDCOM is number of common - 1; LCDSEG is number of segment - 1

Frame rate is the frequency of frame pulse signal (FP).

In general, for mono display, frame rate is approximately 60Hz. For gray level or color display, frame rate might be about 180~90Hz.

For example:

In 4096 colors mode with 160X160 resolution and 48MHz of the system clock

We obtain:

$$48000000 / 160 / 160 / 180 - 2 = 8.4$$

Therefore, LCDCLK [9:0] can be set as 8 or 9 in decimal.

$$\text{Frame rate} = 48000000 / 160 / 160 / (8 + 2) = 187.5\text{Hz (if LCDCLK[9:0]=8)}$$

$$\text{Frame rate} = 48000000 / 160 / 160 / (9 + 2) = 170.5\text{Hz (if LCDCLK[9:0]=9)}$$

In mono mode with 320x240 resolution and 24MHz of the system clock.

We obtain:

$$24000000 / 320 / 240 / 60 - 2 = 3.2$$

Therefore, LCDCLK [9:0] can be set as 3 or 4, even 2 in decimal for higher frame rate.

$$\text{Frame rate} = 24000000 / 320 / 240 / (2 + 2) = 78.1 \text{ Hz (if LCDCLK[9:0]=2)}$$

$$\text{Frame rate} = 24000000 / 320 / 240 / (3 + 2) = 62.5 \text{ Hz (if LCDCLK[9:0]=3)}$$

$$\text{Frame rate} = 24000000 / 320 / 240 / (4 + 2) = 52 \text{ Hz (if LCDCLK[9:0]=4)}$$

Important Note: To decrease power consumption and increase CPU performance on external memory device, system designers should try to minimize LCD frame rate as short as possible until there is no flicking phenomenon on a LCD panel.

P_LCD_Segment		0x7982								LCD Segment Number Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	LCDSEG								
Default		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	LCDSEG	R/W	LCD Panel Segment Number Register LCDSEG [3:0] is fixed to "1111" in binary.	Range= 15 ~ 319 LCDSEG[3:0] should be "1111".

This register defines the segment number of LCD panel that equals to LCDSEG [8:0] +1. It must be the multiple of 16. The maximum segment number is 320 and minimum segment number is 16.

Therefore, LCDSEG [3:0] should be equal to "1111" in binary. Attempt to write "0" to any of LCDSEG [3:0] will be in vain. In other words, LCDSEG [3:0] is read only.

P_LCD_Common		0x7983							LCD Common Number Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	LCDCOM								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	LCDCOM	R/W	LCD Panel Common Number Register.	Range= 0~319

LCDCOM [8:0] control register defines the vertical size of aLCD panel. The actual size (number of common) is LCDCOM [8:0] +1. The maximum common number is 320 and minimum common number is 1.

P_LCD_Buffer_LowAdr					LCD Buffer Address A15 ~ A0												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	LCDBUFAL																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:0]	LCDBUFAL	R/W	LCD Buffer Address [15:0]	

P_LCD_Buffer_HighAdr 0x7985							LCD Buffer Address A25~A16									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	LCDBUFAH									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

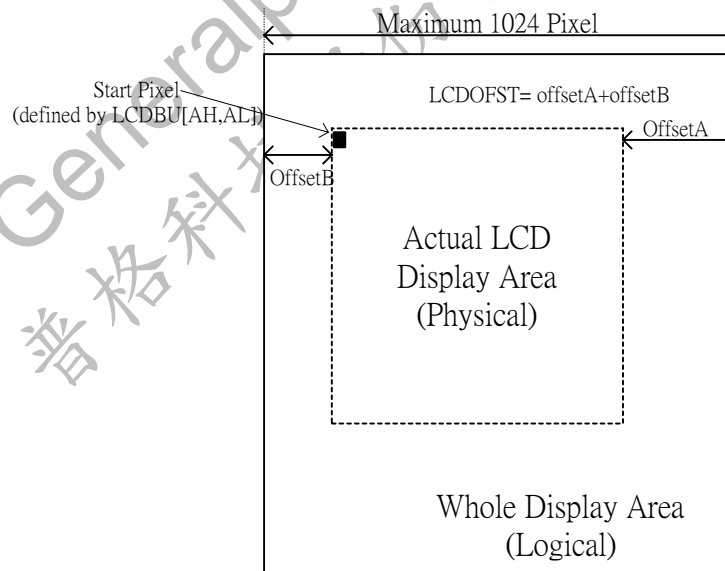
Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:0]	LCDBUFAH	R/W	LCD Buffer Address [25:16]	

LCDBUFAH and LCDBUFAL construct a 26-bit addressing register to define the start address of a LCD buffer.

P_LCD_Buffer_Offset						0x7986											LCD Offset Size										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Function	-	-	-	-	-	-	LCDOFST																				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0											

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LCDOFST	R/W	LCD Virtual Page Offset (Unit: pixel). The value given here must be the multiple of 16; that is, the bit3~bit0 must be all 0s.	

Note: $LCDOFST + LCDSEG + 1$ should smaller than 1024



P_LCD_Buffer_Offset and P_LCD_Buffer_Segment control registers set up the horizontal size of a virtual page. Note that the maximum virtual page is 1024 pixels. To move the actual display area horizontally or vertically, programmers should modify P_LCD_Buffer_HighAdr and P_LCD_Buffer_LowAdr control registers.

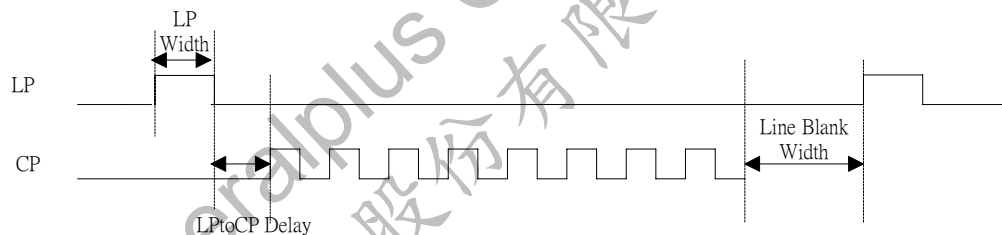
P_LCD_Timing_Ctrl 0x7987
LCD Control Signal Timing Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	LBVL				LPW				LPCPD			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:8]	LBVL	R/W	Line Blank Width Please refer to following timing diagram.	$T = (LBVL + 1) \times CLCPCLK$ Range= 0~15
[7:4]	LPW	R/W	LP Pulse Width Please refer to following timing diagram.	$T = (LPW + 1) \times CLCPCLK$ Range= 0~15
[3:0]	LPCPD	R/W	LP to CP Delay Please refer to following timing diagram.	$T = (LPCPD + 1) \times CLCPCLK$ Range= 0~15

Note: Generally, this register does not need to be changed. The purpose of this register is to adjust the LCD control signals for special LCD drivers.

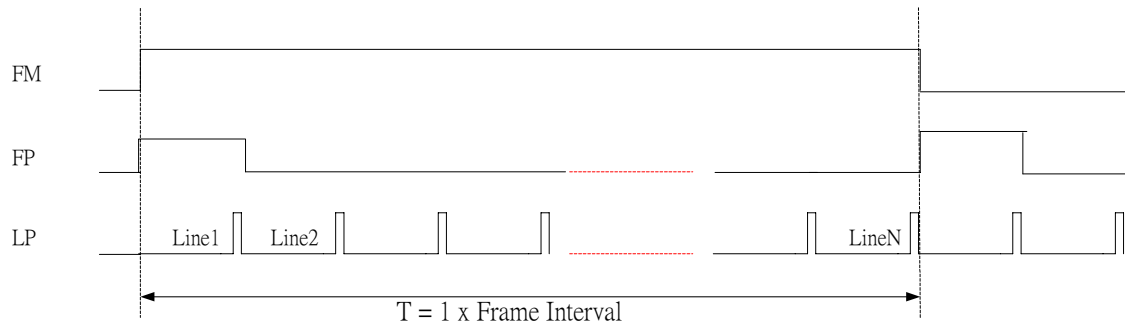
$CLCPCLK = LCDCLK[9:0] + 2$


P_LCD_Frame_Ctrl 0x7988
LCD Frame Modulation Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BCMOD	-	-	-	-	-	-	-	MVAL							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	BCMOD	R/W	LCD Frame Modulation Type. Please refer to the following diagram.	0= B type (Typical) 1= C type
[14:8]			Reserved	
[7:0]	MVAL	R/W	Define the frequency of frame modulation when C type is active.	Range= 0 ~ 255

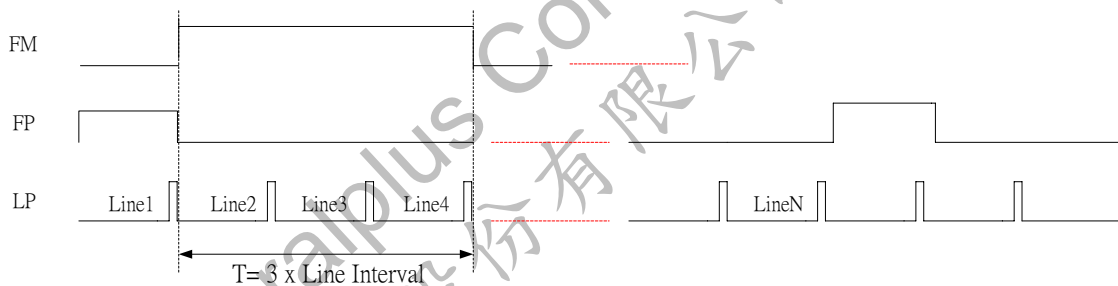
For B Type Frame Modulation Mode (BCMOD=0), FM changes its state for each FP signal.



For C type frame modulation mode (BCDMOD=1), FM changes its state for each (MAL [7:0] +1) LP signal.

For example, if MVAL [7:0] =2, FM signals changes its state each three LP signals. See the following diagram for reference.

If MVAL [7:0] +1 equals the number of common, the C type FM signal is the same as B type FM signal.



P_LCD_Palette_Ctrl 0x7989

LCD Palette Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	OVIF/C	-	-	-	-	-	-	-	BPR	-	-	-	BPP			LCDBW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	OVIF/C	R/W	LCD DMA operations overflow. This bit is set to "1" by hardware if the LCD DMA operation overflow is asserted.	Read 0 = Not Occurs Read 1 = Happened Write 0 = No effect Write 1 = Clear the flag
[14:8]			Reserved	
7	BPR	R/W	Bypass Palette Register Setup. This control bit can be set to "1" only when 1-bit-per-pixel or 12-bit-per-pixel configuration is selected (BPP=000).	0= Not Bypass (Palette is active) 1= Bypass (Palette is not active)

[6:4]			Reserved	
[3:1]	BPP	R/W	Bit Per Pixel Configuration on LCD Buffer.	000= 1 bit/pixel 001= 2 bits/pixel 010= 4 bits/pixel 011= 8 bits/pixel 100= 12 bits/pixel 101~111= Reserved
0	LCDBW	R/W	B/W mode and color mode.	0= Color Mode 1= B/W Mode(including gray mode)

Palette is a look-up table defining the relationship between data in LCD buffer and data to be displayed. Using 2-bit-per-pixel configuration as an example, the 2-bit-per-pixel means one pixel requires two bits memory in a LCD buffer; therefore, there are up to 4 colors or 4 gray levels in each display frame. However, the two bits data cannot define the actual pattern to be displayed. Instead, the 2-bit-data is just an index to the palette. Palette registers define the actual display pattern corresponding to 2-bit-index, "00", "01", "10" and "11". In other words, for 2-bit-per-pixel configuration, it takes 4 palette registers; meaning look-up table depth is 4. The LCD controller will perform this look-up table operation by hardware automatically.

	Valid Palette Depth	Palette Register Address
1 bit-per-pixel	2	0x7A00 ~ 0x7A01
2 bit-per-pixel	4	0x7A00 ~ 0x7A03
4 bit-per-pixel	16	0x7A00 ~ 0x7A0F
8 bit-per-pixel	256	0x7A00 ~ 0x7AFF

BPR control bit (bypass Palette table) is valid only when 1-bit-per-pixel and B/W mode is selected. In most cases of 1-bit-per-pixel configuration, it is mono display mode and Palette table is not used. Therefore, Palette function can be bypassed by setting BPR to "1". However, programmers can still use palette function on 1-bit-per-pixel mode (BPR="0"). In this case, GPL162002A/162003A LCD controller will read one index bit from LCD buffer, next look up the corresponding first two palette registers, and then finally send the corresponding gray level or color information to external LCD driver.

In the case of 12-bit-per-pixel (4096 colors solution), GPL162002A/162003A does not support Palette function. In other words, when 12-bit-per-pixel configuration is selected, Palette function would be bypassed by hardware (BPR="1").

LCDBW and BPP are used to set up the pixel configuration. If LCDBW is set to "1", Black & White mode is enabled. In this mode, it supports mono, 4-gray-level, and 16-gray-level types for each pixel. These

three types are corresponding to 1-bit-per-pixel, 2-bit-per-pixel and 4-bit-per-pixel respectively. Note that Black& White mode does not support neither 8-bit-per-pixel nor 12-bit-per-pixel configuration. That is, BPP cannot be set to neither “011” nor “100” if LCDBW is equal to “1”.

Following diagram depicts the configuration that GPL162002A/162003A supports and not supports.

	Bypass Palette B/W Mode (LCDBW=1)	Palette Valid B/W Mode (LCDBW=1)	Bypass Palette Color Mode (LCDBW=0)	Palette Valid Color Mode (LCDBW=0)
1-bit-per-pixel	O*	O*	X	O
2-bit-per-pixel	X	O	X	O
4-bit-per-pixel	X	O	X	O
8-bit-per-pixel	X	X	X	O
12-bit-per-pixel	X	X	O	X

O: Supported, X: Not supported

*Note that BPR control bit is used to determine whether to bypass palette or not ONLY when 1-bit-per-pixel mode is selected.

Note: If LCDBW is cleared to “0”, color mode is enabled. Color mode can only support 4-bit and 8-bit data buses. Simply, if color mode is enabled (LCDBW=0), BUSW cannot be “00”.

P_LCD_Attri_Ctrl 0x798A LCD Attribute Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	VerINV	HORINV	DATAINV	NEGFIL
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	VERINV	R/W	Vertical Invert in LCD Display	0= Disable 1= Enable
2	HORINV	R/W	Horizontal Invert in LCD Display	0= Disable 1= Enable
1	DATAINV	R/W	LCD Display Data Order Reversed D[7:0] => D[0:7]	0= Disable 1= Enable
0	NEGFILM	R/W	LCD Data Exclusive 0xFF to Display 0x55 => 0xAA	0= Disable 1= Enable

11.6 Operation during Wait/Halt/Standby & Wakeup Procedure

In wait mode, the LCD is able to remain functioning even if CPU is turned off because GPL162002A/162003A keeps PLL clock activating. On the other hand, the PLL clock is shut down in both halt and standby modes that will cause LCD unable to display in such modes. Therefore,

Generalplus recommends set the LCDEN control bit to “0” before entering halt mode and standby mode. After GPL162002A/162003A wakes up from those modes, set the LCDEN control bit to “1” again. Simply, programmers are able to turn on/off LCD panel via the configuration of LCDEN control bit. The GPL162002A/162003A will automatically perform the LCD power on/off procedures in order to avoid unexpected lines occurring on the LCD panel.

11.7 LCD Image Resource File Tooling

Generalplus offers some PC tools to transfer BMP file to the LCD buffer format. Tools are as follows: PO4COLOR.EXE, SWAPWORD.EXE, BMP2GIM.EXE and POBMP.EXE.

<256 Color BMP File> to <1-Bit-Per-Pixel Format of LCD Buffer>(2 Mono)

Conversion Batch File

```
po bmp /f2 %1.bmp /d /o
swapword %1.put %.raw /o /b4
```

Note: there is no header in front of the .raw file

<256 Color BMP File> to <4-Bit-Per-Pixel Format of LCD Buffer>(16 Gray)

Conversion Batch File

```
po4color b %1.bmp %1.bin /o /b-4 /x
swapword %1.bin %1.raw /o /b4
```

Note: The first word of .raw represents the number of picture width and the second word of .raw means the number of picture height. The image raw data starts from third word.

<256 Color BMP File> to <8-Bit-Per-Pixel Format of LCD Buffer>(256 Color)

Conversion Batch File

```
bmp2gim %.bmp %.GIM
```

Note: The first eight words of .gim are reserved for header signature. The following 256 words are the information for LCD Palette, and the rest of data are the real image raw data

<High Color (24-bit) BMP File> to <12-Bit-Per-Pixel Format of LCD Buffer> (4096 Color)

Conversion Batch File

```
po4color e %1.bmp %1.bin /o /b-4 /t0
swapword %1.bin %1.raw /o /b4
```

Note: The first word of .raw represents the number of picture width and second word of .raw represents the number of picture height. The image raw data starts from third word.

11.8 Program Examples

160 (Segment) x 160 (Common) 256 color, 4-bit interface LCD buffer start address = 0x004_0000

```
.DEFINE          SEGMENT          160
.DEFINE          COMMON           160

r1 = (48000000/SEGMENT/COMMON/180-2)
[P_LCD_Clock] = r1
r1 = 4
[P_LCD_Buffer_HighAdr] = r1
r1 = 0
[P_LCD_Buffer_LowAdr] = r1
[P_LCD_Buffer_Offset] = r1
[P_LCD_Timing_Ctrl] = r1
[P_LCD_Frame_Ctrl] = r1

r1 = SEGMENT -1
[P_LCD_Segment] = r1
r1 = COMMON -1
[P_LCD_Common] = r1

r1 = 0x0006 // 8-bit-per-pixel (256 Colors)
[P_LCD_Palette_Ctrl] = r1
r1 = 0x2100 // Data width 4-bit
[P_LCD_Ctrl] = r1

call F_Fill256CPalette
.....
.....
```

F_Fill256CPalette:


```
r1 = [P_LCD_Ctrl]
r1 = r1 & (~0x2000)           // Clear LCDEN bit to 0
[P_LCD_Ctrl] =r1
call F_DelayOneFrameInterval // Delay one frame interval
.....                       // Fill palette from 0x7A00 to 0x7AFF
r1 = [P_LCD_Ctrl]
r1 = r1 | (0x2000)           // Set LCDEN bit to 1
[P_LCD_Ctrl] =r1
retf
```

Note that palette can be modified ONLY when LCDEN control bit is cleared to 0 (P_LCD_Ctrl.bit13=0).

Copy 320x240 256-Color Information to both LCD Palette and LCD Buffer

////////////////////////////////////

FG_PutImage320x240x256C:

////////////////////////////////////

```

                                r1 = [R_BitmapIndex]
                                r2 = T_BMPStartAddressTable
                                r2 += r1
                                r2 = [r2]
                                r1 = [r2++]
                                r2 = [r2]

                                r2 = r2 lsl 4
                                r2 = r2 lsl 4
                                r2 = r2 lsl 2
                                ds = r2                                // Setup Data Segment

                                r1 = r1 + 8
                                r4=P_LCD_Palette
                                // Fill LCD Palette
L_FillPalette?:
                                r3=DS:[r1++]
                                [r4++]=r3
                                cmp r4,P_LCD_Palette+256
                                jne L_FillPalette?

                                r2 = 0x0000
                                r4=ds                                // Fill LCD Buffer at 0x3_0000
                                ds=r4
L_FillLCDBuffer?:
                                r3 = DS:[r1++]
                                r4=ds
                                ds=3
                                DS:[r2++] = r3
                                cmp r2,(320*240*8/16)
                                jne L_FillLCDBuffer?

                                retf;

```

12 TFT LCD

12.1 Introduction

GPL162003A does not have TFT LCD feature, so GPL162003A programmer can ignore this chapter.

GPL162002A provides a TFT LCD controller supporting data types of parallelRGB (5-6-5), serial delta RGB, serial stripe RGB, serial YUV, serial YCbCr, and CCIR656. The maximum horizontal resolution of TFT controller reaches 640 pixels, and the maximum vertical resolution of TFT controller reaches 480 pixels. For the resolution setting, it could be programmed by giving designated values to internal registers. The TFT controller mainly provides four timing signals and an 8-/ 16-bit data signal to control external TFT module. These are VSYNC, HSYNC, DE, DCLK and DATA. Besides, GPL162002A provides a special function, PIP, which can overlap the main display window to create up to 4 sub display windows.

Feature

- Maximum horizontal and vertical pixels are 640 X 480.
- The TFT clock is divided from the system clock (1, 1/2, 1/4, 1/8 of SYSCLK).
- The width and polarity of HSYNC and VSYNC can be programmed.
- Active region and blank region clock and the TFT clock of the horizontal pixels are programmable.
- Active region and blank region clock and the TFT clock of the vertical line are programmable.
- Support parallel RGB mode (R:5-bit, G:6-bit, B: 5-bit).
- Support delta-RGB/ stripe-RGB mode.
- Support the adjustable order of RGB data for each odd and even line in RGB mode.
- Support YUV or YCbCr mode.
- Support the adjustable order of YUV/YCbCr data for each line in YUV mode.
- Support CCIR656 mode.
- Support four sub frame buffers reading by PIP DMA.
- The scrolling function is supported for each PIP module.

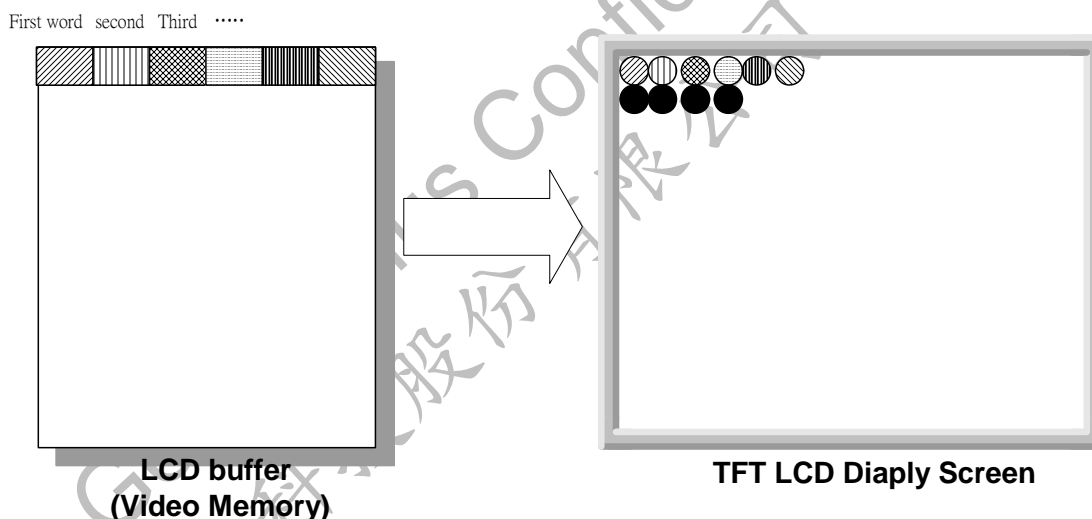
12.2 TFT Interface Signal

Signal	I/O	Description
TFT_Vsync	O	TFT vertical synchronous signal (shared with GPIO portC3)
TFT_Hsync	O	TFT horizontal synchronous signal (shared with GPIO portC2)
TFT_DE	O	TFT data enable (shared with GPIO portC1)
TFT_CLK	O	TFT data clock (shared with GPIO portC0)
TFT_DATA	O	TFT data bus, 8-bit for serial/ 16-bit for parallel. (shared with GPIO portA[15:0])

12.3 TFT LCD Buffer

In GPL162002A the control registers, P_TFT_DMAStart_AH (0x7D0D) and P_TFT_DMAStart_AL (0x7D0E), are to define TFT LCD buffer start address with the 80M-word addressing field as a LCD buffer. Certainly, the area, defined as a LCD buffer, can be SRAM, ROM, or Flash. LCD buffer can also be internal RAM.

LCD buffer size definition is implied in two control registers, P_TFT_Pixel_Num and P_TFT_Line_Num. The maximum number of common lines (horizontal line) to be supported is 640 pixels and the maximum segment line (vertical line) is 480 pixels. GPL162002A TFT LCD can support up to 65536-color display with 16 bits representing a pixel. This 16-bit-per pixel configuration is composed of R-5 bit, G-6 bit, and B-5 bit. In most cases, the size of LCD buffer equals to $\{(P_TFT_Pixel_Num + 1) \times (P_TFT_Line_Num + 1) \times 16\}$ bits.



12.4 Sub Frame Display (PIP)

GPL162002A support four sub frame display function. The main display frame can be covered with sub frames of their own data buffer without changing original main frame data in LCD buffer. This is a very useful function for game or animation display. If these four sub frames overlap with each other simultaneously, the priority is as PIP3>PIP2>PIP1>PIP0.

Each PIP control supports scrolling function; this means, the sub frame buffer data are larger than data of sub frame display screen size. By this scrolling method, it is easy to review all sub frame buffer data for each PIP.

12.5 Control Register

TFT LCD Control Register Summary Table

Name	Address	Description
P_TFT_CTRL	0x7D00	TFT Control Register
P_TFT_DCLK_CTRL	0x7D01	TFT Data Clock Control Register
P_TFT_INT_CTRL	0x7D02	TFT Interrupt Control Register
P_TFT_H_WIDTH	0x7D03	TFT Horizontal Width
P_TFT_H_START	0x7D04	TFT Horizontal Start Location
P_TFT_H_END	0x7D05	TFT Horizontal End Location
P_TFT_HSYNC_SETUP	0x7D06	TFT Hsync Setup Register
P_TFT_V_WIDTH	0x7D07	TFT Vertical Width
P_TFT_V_START	0x7D08	TFT Vertical Start Location
P_TFT_V_END	0x7D09	TFT Vertical End Location
P_TFT_VSYNC_SETUP	0x7D0A	TFT Vsync Setup Register
P_TFT_RGB_CTRL	0x7D0B	TFT RGB Mode Control Register
P_TFT_YUV_CTRL	0x7D0C	TFT YUV Mode Control Register
P_TFT_DMASTART_AH	0x7D0D	TFT DMA Start High Address
P_TFT_DMASTART_AL	0x7D0E	TFT DMA Start Low Address
P_TFT_DMA_OFFSET	0x7D0F	TFT DMA Offset Address
P_TFT_PIXEL_NUM	0x7D10	TFT Pixel Numbers in Each Line
P_TFT_LINE_NUM	0x7D11	TFT Line Number in Each Frame
P_TFT_PIP0_CTRL	0x7D12	TFT PIP0 Control Register
P_TFT_PIP0_VIR_SAH	0x7D13	TFT PIP0 Virtual Frame Buffer Start High Address
P_TFT_PIP0_VIR_SAL	0x7D14	TFT PIP0 Virtual Frame Buffer Start Low Address
P_TFT_PIP0_VIR_EAH	0x7D15	TFT PIP0 Virtual Frame Buffer End High Address
P_TFT_PIP0_VIR_EAL	0x7D16	TFT PIP0 Virtual Frame Buffer End Low Address
P_TFT_PIP0_STARTAH	0x7D17	TFT PIP0 Frame Buffer Start High Address
P_TFT_PIP0_STARTAL	0x7D18	TFT PIP0 Frame Buffer Start Low Address
P_TFT_PIP0_H_START	0x7D19	TFT PIP0 Horizontal Start Location in Each Line
P_TFT_PIP0_H_END	0x7D1A	TFT PIP0 Horizontal End Location in Each Line
P_TFT_PIP0_V_START	0x7D1B	TFT PIP0 Vertical Start Location in Each Frame
P_TFT_PIP0_V_END	0x7D1C	TFT PIP0 Vertical End Location in Each Frame
P_TFT_PIP1_CTRL	0x7D1D	TFT PIP1 Control Register
P_TFT_PIP1_VIR_SAH	0x7D1E	TFT PIP1 Virtual Frame Buffer Start High Address
P_TFT_PIP1_VIR_SAL	0x7D1F	TFT PIP1 Virtual Frame Buffer Start Low Address
P_TFT_PIP1_VIR_EAH	0x7D20	TFT PIP1 Virtual Frame Buffer End High Address
P_TFT_PIP1_VIR_EAL	0x7D21	TFT PIP1 Virtual Frame Buffer End Low Address
P_TFT_PIP1_STARTAH	0x7D22	TFT PIP1 Frame Buffer Start High Address
P_TFT_PIP1_STARTAL	0x7D23	TFT PIP1 Frame Buffer Start Low Address
P_TFT_PIP1_H_START	0x7D24	TFT PIP1 Horizontal Start Location in Each Line
P_TFT_PIP1_H_END	0x7D25	TFT PIP1 Horizontal End Location in Each Line

Name	Address	Description
P_TFT_PIP1_V_START	0x7D26	TFT PIP1 Vertical Start Location in Each Frame
P_TFT_PIP1_V_END	0x7D27	TFT PIP1 Vertical End Location in Each Frame
P_TFT_PIP2_CTRL	0x7D28	TFT PIP2 Control Register
P_TFT_PIP2_VIR_SAH	0x7D29	TFT PIP2 Virtual Frame Buffer Start High Address
P_TFT_PIP2_VIR_SAL	0x7D2A	TFT PIP2 Virtual Frame Buffer Start Low Address
P_TFT_PIP2_VIR_EAH	0x7D2B	TFT PIP2 Virtual Frame Buffer End High Address
P_TFT_PIP2_VIR_EAL	0x7D2C	TFT PIP2 Virtual Frame Buffer End Low Address
P_TFT_PIP2_STARTAH	0x7D2D	TFT PIP2 Frame Buffer Start High Address
P_TFT_PIP2_STARTAL	0x7D2E	TFT PIP2 Frame Buffer Start Low Address
P_TFT_PIP2_H_START	0x7D2F	TFT PIP2 Horizontal Start Location in Each Line
P_TFT_PIP2_H_END	0x7D30	TFT PIP2 Horizontal End Location in Each Line
P_TFT_PIP2_V_START	0x7D31	TFT PIP2 Vertical Start Location in Each Frame
P_TFT_PIP2_V_END	0x7D32	TFT PIP2 Vertical End Location in Each Frame
P_TFT_PIP3_CTRL	0x7D33	TFT PIP3 Control Register
P_TFT_PIP3_VIR_SAH	0x7D34	TFT PIP3 Virtual Frame Buffer Start High Address
P_TFT_PIP3_VIR_SAL	0x7D35	TFT PIP3 Virtual Frame Buffer Start Low Address
P_TFT_PIP3_VIR_EAH	0x7D36	TFT PIP3 Virtual Frame Buffer End High Address
P_TFT_PIP3_VIR_EAL	0x7D37	TFT PIP3 Virtual Frame Buffer End Low Address
P_TFT_PIP3_STARTAH	0x7D38	TFT PIP3 Frame Buffer Start High Address
P_TFT_PIP3_STARTAL	0x7D39	TFT PIP3 Frame Buffer Start Low Address
P_TFT_PIP3_H_START	0x7D3A	TFT PIP3 Horizontal Start Location in Each Line
P_TFT_PIP3_H_END	0x7D3B	TFT PIP3 Horizontal End Location in Each Line
P_TFT_PIP3_V_START	0x7D3C	TFT PIP3 Vertical Start Location in Each Frame
P_TFT_PIP3_V_END	0x7D3D	TFT PIP3 Vertical End Location in Each Frame

P_TFT_CTRL
0x7D00
TFT Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TFTEN	-	-	-	VS_TYPE	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TFTEN	R/W	TFT Controller Enable. Set this bit to 1 to enable TFT interface. Clear to 0, then TFT interface pin will reinstate as GOIP.	0: Disable 1: Enable
[14:12]			Reserved	
11	VS_TYPE	R/W	Vertical Synchronous Type Control. If this bit set to 1, the odd field and even field of vertical will add half-line additionally.	0: Not add 1: add half-line
[10:0]			Reserved	

P_TFT_DCLK_CTRL								TFT Data Clock Control Register								
0x7D01																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	DCLK_INV	-	-	-	DCLK_SEL				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
8	DCLK_INV	R/W	TFT data Clock Inverse. If this bit is set to 1, it will latch data by negative TFT_CLK edge; otherwise it will latch data by positive TFT_CLK edge.	0: Not inverse (positive edge) 1: Inverse (negative edge)
[7:5]			Reserved	
[4:0]	DCLK_SEL	R/W	TFT Clock Selection. If these bits set to all "0"s, then the TFT_CLK is equal to system clock. If these are set to all "1"s, then TFT_CLK is equal to system clock/2 and so on.	TFT_CLK = System Clock / (DCLK_SEL+1)

P_TFT_INT_CTRL		0x7D02		TFT Interrupt Control Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	UF_F/C	UF_EN	-	-	FE_F/C	FE_EN	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	UF_F/C	R/W	TFT FIFO Underflow Error Interrupt Flag. This bit is set to 1 by hardware when TFT LCD buffer data are not prompt in transmitting out. Write 1 to clear the flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
14	UF_EN	R/W	TFT FIFO Underflow Error Interrupt Enable. If this bit set to 1, and FIFO underflow interrupt occurs, hardware will issue an IRQ5 or FIQ to CPU. If this bit is clear to 0, this interrupt will be marked. To select between IRQ5 or FIQ. Please refer to Chapter Interrupt .	0: Disable 1: Enable
[13:12]			Reserved	
11	FE_F/C	R/W	Frame End Interrupt Flag. This bit is set to 1 by hardware when TFT complete one frame data transmission. Write 1 to clear the flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
10	FE_EN	R/W	TFT Frame End Interrupt Enable. If this bit set to 1, and frame end interrupt	0: Disable 1: Enable

Bit	Function	Type	Description	Condition
			occurs, hardware will issue an IRQ5 or FIQ to CPU. If this bit is clear to 0, this interrupt will be marked. To select between IRQ5 or FIQ. Please refer to Chapter Interrupt .	
[9:0]			Reserved	

P_TFT_H_WIDTH
0x7D03
TFT Horizontal Width

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	H_WIDTH											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	H_WIDTH	R/W	TFT Horizontal Total Clock Width. This register is to set the number of TFT_CLK of one line. See the following diagram for details.	Total horizontal TFT_CLK = H_WIDTH+1

P_TFT_H_START
0x7D04
TFT Horizontal Start Location

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	H_START										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	H_START	R/W	TFT Horizontal Display Start Location. This register is to set the number of TFT_CLK of TFT_Hsync to first horizontal pixel. See the following diagram for details.	Horizontal start location = H_START+1

P_TFT_H_END
0x7D05
TFT Horizontal End Location

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	H_END											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	H_END	R/W	TFT Horizontal Display End Location. This register is to set the number of TFT_CLK of	Horizontal end location = H_END+1

Bit	Function	Type	Description	Condition
			TFT_Hsync to last horizontal pixel. See the following diagram for details.	

P_TFT_HSYNC_SETUP
0x7D06
TFT Hsync Setup Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HS_POL	-	-	-	-	HS_WIDTH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	HS_POL	R/W	TFT Horizontal Synchronous Polarity	0: Negative (Low pulse) 1: Positive (High pulse)
[14:11]			Reserved	
[10:0]	HS_WIDTH	R/W	TFT Horizontal Pixel Width This register is to set the number of TFT_CLK of horizontal pixel. See the following diagram for details.	Horizontal pixel width = HS_WIDTH+1

P_TFT_V_WIDTH
0x7D07
TFT Vertical Width

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	V_WIDTH									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	V_WIDTH	R/W	TFT Vertical Total line. This register is to set the total vertical width of one frame. See the following diagram for details.	Total vertical line = V_WIDTH+1

P_TFT_V_START
0x7D08
TFT Vertical Start Location

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	V_START								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
[8:0]	V_START	R/W	TFT Horizontal Display Start Line. This register is to set the number of TFT_Vsync to first meaningful TFT_Hsync. See the following diagram for details.	Vertical start line = V_START+1

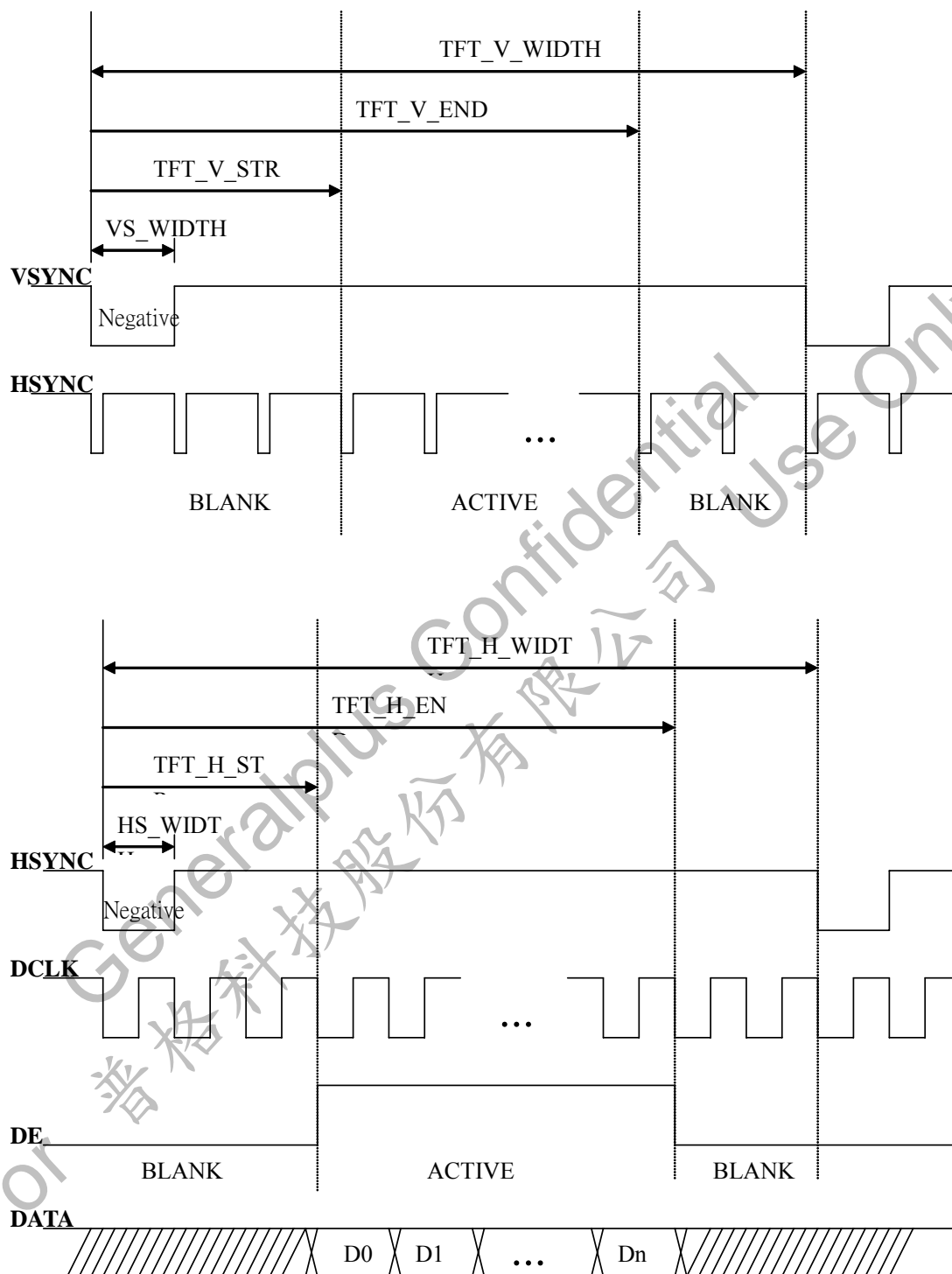
P_TFT_V_END		0x7D09						TFT Vertical End Location									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	V_END									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	V_END	R/W	TFT Vertical Display End Line. This register is to set the number of TFT_Vsync to last meaningful TFT_Hsync. See the following diagram for details.	Horizontal end location = V_END+1

P_TFT_VSYNC_SETUP		0x7D0A							TFT Vsync Setup Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function		VS_POL	-	-	-	-	-	-			VS_WIDTH							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	VS_POL	R/W	TFT Vertical Synchronous Polarity.	0: Negative (Low pulse) 1: Positive (High pulse)
[14:9]			Reserved	
[8:0]	VS_WIDTH	R/W	TFT Vertical Line Width. This register is to set the number of real used vertical line. See the following diagram for details.	Vertical used line width = VS_WIDTH+1

TFT Timing



P_TFT_RGB_CTRL			0x7D0B			TFT RGB Mode Control Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RGB_M	RGB_DMEN	-	-	-	-	-	-	-	ODD_L_TYPE			-	EVEN_L_TYPE		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RGB_M	R/W	RGB Parallel / Serial Mode Selection. In parallel mode, IOA [15:0] is used as TFT data bus. In this mode, the keyscan function is invalid. In serial mode, IOA [7:0] is used as TFT data. In this mode, the keyscan function can be active with TFT at the same time.	0: Parallel (16 bit) 1: Serial (8 bit)
14	RGB_DMEN	R/W	RGB With Dummy Data Enable. This bit is used for RGB serial mode only. If this bit is set to 1, a pixel takes 4 TFT_CLK, which includes R-G-B clocks and a dummy data clock.	0: Disable 1: Enable
[13:7]			Reserved	
[6:4]	ODD_L_TYPE	R/W	Odd line Serial RGB Data Arrangement. These bits are used for RGB serial mode only.	000: RGB 001: RBG 010: GRB 011: GBR 100: BRG 101: BGR 110~111: Reserved
3			Reserved	
[2:0]	EVEN_L_TYPE	R/W	Even line Serial RGB Data Arrangement. These bits are used for RGB serial mode only.	000: RGB 001: RBG 010: GRB 011: GBR 100: BRG 101: BGR 110~111: Reserved

P_TFT_YUV_CTRL			0x7D0C			TFT YUV Mode Control Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	YUV_EN	YUV_M	CCIR656_EN	-	-	-	-	-	-	-	-	SHARE	-	-	YUV_TYPE	
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
15	YUV_EN	R/W	YUV Mode Enable.	0: Disable 1: Enable
14	YUV_M	R/W	YCbCr and YUV Selection. This bit is valid only when YUV is enabled.	0: YCbCr 1: YUV
13	CCIR656_EN	R/W	CCIR656 Mode Enable. This bit is valid only when YUV is enabled.	0: Disable 1: Enable
[12:5]			Reserved	
4	SHARE	R/W	RGB Data Share Enable. If this bit is set to 1, then there is one RGB data for U (Cb) YV (cr) Y. Otherwise, there are two RGB data for U (Cb) YV (Cr) Y. This bit is valid only when YUV is enabled.	0: Not share 1: Share
[3:2]			Reserved	
[1:0]	YUV_TYPE	R/W	YU(Cb)V(Cr) data Arrangement. These bits are valid only when YUV is enabled.	00: U(Cb)YV(Cr)Y 01: V(Cr)YU(Cb)Y 10: YU(Cb)YV(Cr) 11: YV(Cr)YU(Cb)

P_TFT_DMASTART_AH 0x7D0D TFT DMA Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	DMA_SAH	R/W	TFT DMA Start Address. This register is to set TFT LCD buffer start address high byte.	

P_TFT_DMASTART_AL 0x7D0E TFT DMA Start Low Address

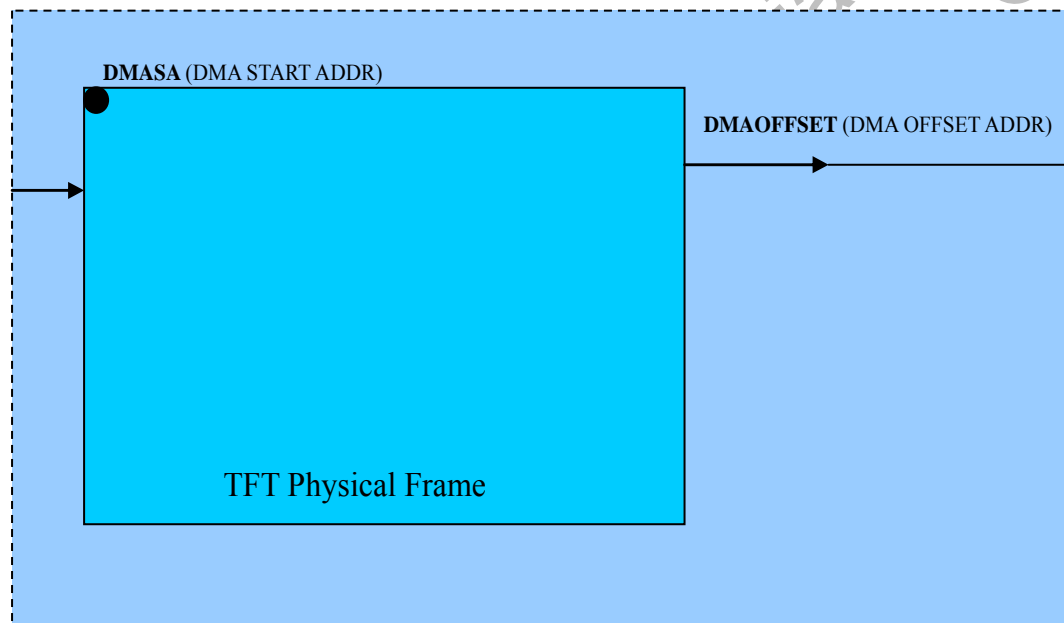
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DMA_SAL	R/W	TFT DMA Start Address. This register is to set TFT LCD buffer start address low byte.	

P_TFT_DM_OFFSET					0x7D0F					TFT DMA Offset Address						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	DMA_OFFSET										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	DMA_OFFSET	R/W	TFT DMA Address Offset For Each Line.	

TFT Virtual Frame



P_TFT_DM_OFFSET and P_TFT_PIXEL_NUM control registers set up the horizontal size of virtual page. The maximum virtual page is of 1024 pixels. To move the actual display area horizontally or vertically, users can just modify P_TFT_DMASTART_AH and P_TFT_DMASTART_AL control registers.

P_TFT_PIXEL_NUM		0x7D10						TFT Pixel Numbers in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIXEL_NUM										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	PIXEL_NUM	R/W	The number of pixel in each line.	TFT line pixel =

			The maximum horizontal pixel number on GPL162002A is 640.	PIXEL_NUM+1
--	--	--	---	-------------

P_TFT_LINE_NUM		0x7D11						TFT Line Numbers in Each Line									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	LINE_NUM									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LINE_NUM	R/W	The number of line in each frame. The maximum vertical line number on GPL12002A is 480.	TFT frame line = LINE_NUM+1

PIP Special Function (Display Priority: PIP3 > PIP2 > PIP1 > PIP0 > Main)

P_TFT_PIP0_CTRL		0x7D12						TFT PIP0 Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0EN	PIP0SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_CTRL		0x7D1D						TFT PIP1 Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1EN	PIP1SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_CTRL		0x7D28																TFT PIP2 Control Register															
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Function		PIP2EN	PIP2SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-																
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

P_TFT_PIP3_CTRL		0x7D33						TFT PIP3 Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP3EN	PIP3SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	PIP#EN	R/W	The PIP# Frame Enable.	0: Disable 1: Enable
14	PIP#SCREN	R/W	The PIP# Scrolling Function Enable. See the following diagram for details.	0: Disable 1: Enable
[13:0]			Reserved	

P_TFT_PIP0_VIR_SAH 0x7D13 TFT PIP0 Virtual Frame Buffer Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP0_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_SAH 0x7D1E TFT PIP1 Virtual Frame Buffer Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_SAH 0x7D29 TFT PIP2 Virtual Frame Buffer Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_SAH 0x7D34 TFT PIP3 Virtual Frame Buffer Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP3_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	PIP#_VIR_SAH	R/W	PIP# Virtual Start High Address. This register is valid only when PIP#SCREN is set to 1. The virtual frame start address means the real address of data, not TFT PIP# buffer start address. See the following diagram for details.	

P_TFT_PIP0_VIR_SAL 0x7D14 TFT PIP0 Virtual Frame Buffer Start Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_SAL 0x7D1F TFT PIP1 Virtual Frame Buffer Start Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_SAL 0x7D2A TFT PIP2 Virtual Frame Buffer Start Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP2_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_SAL 0x7D35 TFT PIP3 Virtual Frame Buffer Start Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP3_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	PIP#_VIR_SAL	R/W	PIP# Virtual Start Low Address This register is valid only when PIP#SCREN is set to 1. The virtual frame start address means the real address of data, not TFT PIP# buffer start address. See the following diagram for details.	PIP3_VIR_SAL = Real data address + 1

P_TFT_PIP0_VIR_EAH 0x7D15 TFT PIP0 Virtual Frame Buffer End High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP0_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_EAH 0x7D20 TFT PIP1 Virtual Frame Buffer End High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_EAH 0x7D2B TFT PIP2 Virtual Frame Buffer End High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_EAH 0x7D36 TFT PIP3 Virtual Frame Buffer End High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP3_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	PIP#_VIR_EAH	R/W	PIP# Virtual End High Address. This register is valid only when PIP#SCREN is set to 1. The virtual frame End address means the real address of data. See the following diagram for details.	

P_TFT_PIP0_VIR_EAL 0x7D16 TFT PIP0 Virtual Frame Buffer End Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_EAL 0x7D21 TFT PIP1 Virtual Frame Buffer End Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_EAL 0x7D2C TFT PIP2 Virtual Frame Buffer End Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP2_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_EAL 0x7D37 TFT PIP3 Virtual Frame Buffer End Low Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP3_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	PIP#_VIR_EAL	R/W	PIP# Virtual End Low Address. This register is valid only when PIP#SCREN is set to 1. The virtual frame End address means the real address of data. See the following diagram for details.	

P_TFT_PIP0_STARTAH 0x7D17 TFT PIP0 Frame Buffer Start High Address

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP0_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_STARTAH					TFT PIP1 Frame Buffer Start High Address											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_STARTAH					0x7D2D	TFT PIP2 Frame Buffer Start High Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_SA_H										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_STARTAH					TFT PIP3 Frame Buffer Start High Address											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP3_SA_H										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:11]			Reserved	
[10:0]	PIP#_SAH	R/W	PIP# Frame Buffer Start High Address. This register set up TFT PIP# LCD buffer address.	

P_TFT_PIP0_STARTAL					0x7D18					TFT PIP0 Frame Buffer Start Low Address						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_STARTAL 0x7D23								TFT PIP1 Frame Buffer Start Low Address								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_STARTAL					0x7D2E		TFT PIP2 Frame Buffer Start Low Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP2_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_STARTAL					0x7D39		TFT PIP3 Frame Buffer Start Low Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP3_SAL																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:0]	PIP#_SAL	R/W	PIP# Frame Buffer Start Low Address. This register set up TFT PIP# LCD buffer address.	

P_TFT_PIP0_H_START 0x7D19 TFT PIP0 Horizontal Start Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_H_START 0x7D24 TFT PIP1 Horizontal Start Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP1_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_H_START 0x7D2F TFT PIP2 Horizontal Start Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP2_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_H_START 0x7D3A TFT PIP3 Horizontal Start Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP3_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	PIP#_H_STR	R/W	The PIP# frame horizontal start location in the main frame. See the following diagram for details.	

P_TFT_PIP0_H_END 0x7D1A TFT PIP0 Horizontal End Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_H_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_H_END 0x7D25 TFT PIP1 Horizontal End Location in Each Line

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP1_H_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_H_END				0x7D30			TFT PIP2 Horizontal End Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP2_H_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_H_END				0x7D3B			TFT PIP3 Horizontal End Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP3_H_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	PIP#_H_END	R/W	The PIP# frame horizontal end location in the main frame. See the following diagram for details.	

P_TFT_PIP0_V_START				0x7D1B			TFT PIP0 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_V_START							TFT PIP1 Vertical Start Location in Each Frame										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP1_V_STR										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_V_START		0x7D31						TFT PIP2 Vertical Start Location in Each Frame									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	PIP2_V_STR									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_V_START				0x7D3C			TFT PIP3 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP3_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	PIP#_V_STR	R/W	The PIP# frame vertical start location in the main frame. See the following diagram for details.	

P_TFT_PIP0_V_END		0x7D1C						TFT PIP0 Vertical End Location in Each Frame								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_V_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

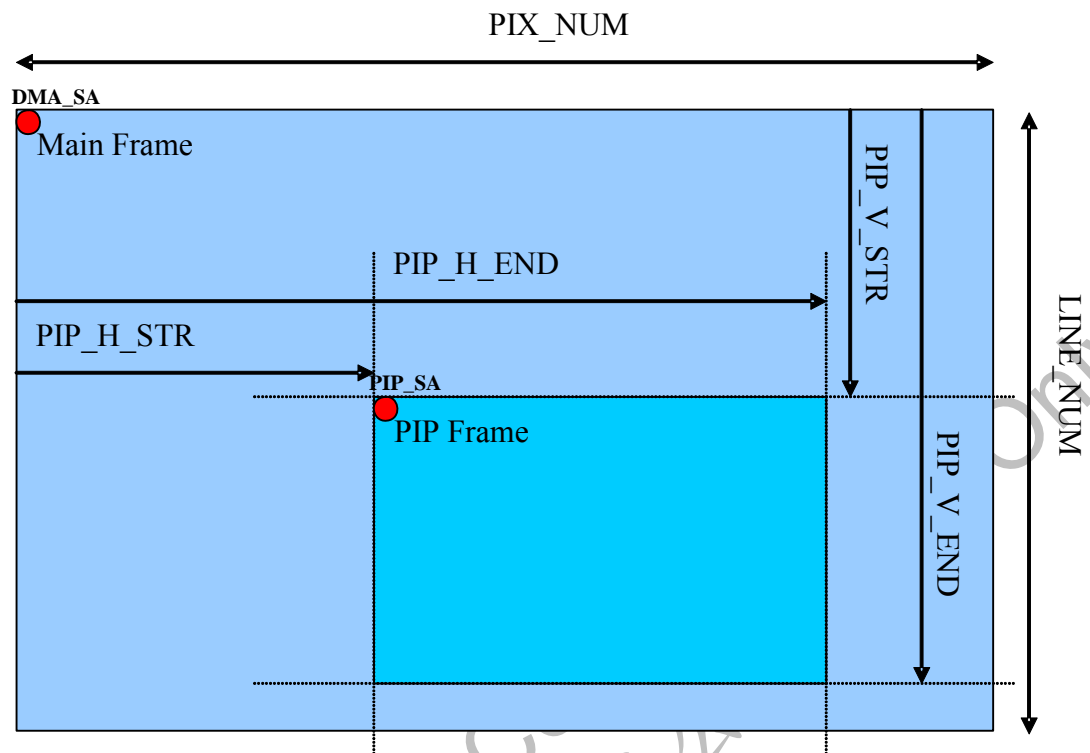
P_TFT_PIP1_V_END			0x7D27				TFT PIP1 Vertical End Location in Each Frame										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP1_V_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_V_END		0x7D32						TFT PIP2 Vertical End Location in Each Frame									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	PIP2_V_END									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

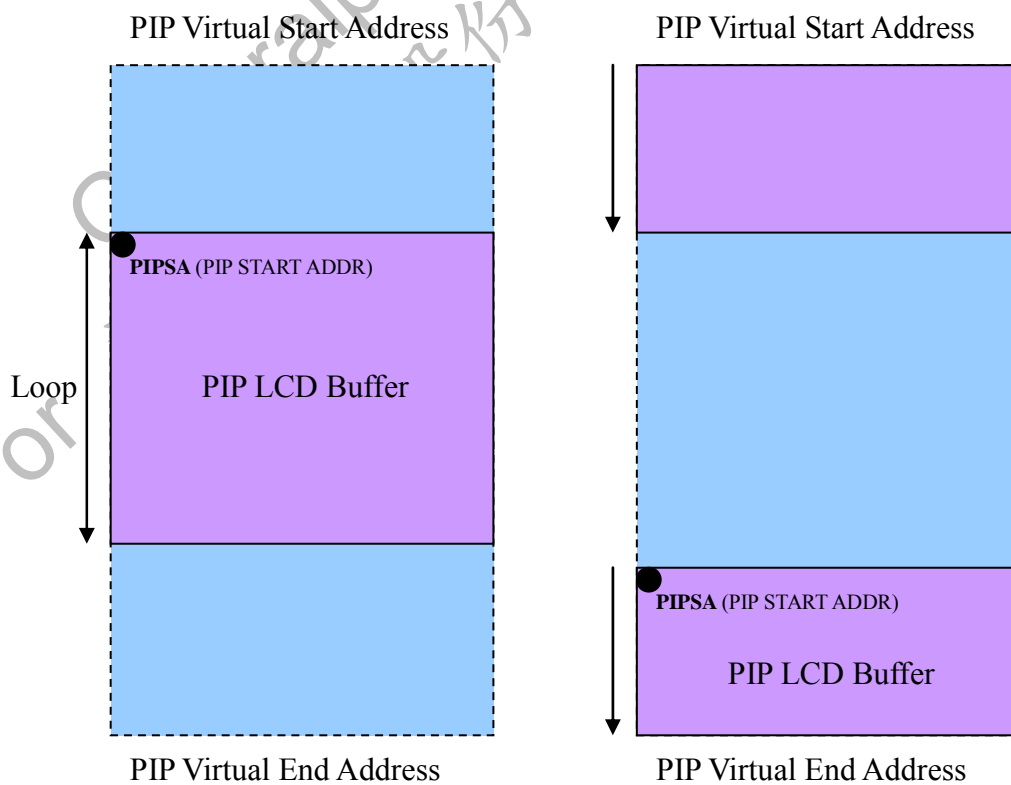
P_TFT_PIP3_V_END			0x7D3D				TFT PIP3 Vertical End Location in Each Frame										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP3_V_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	PIP#_V_END	R/W	The PIP# frame vertical end location in the main frame. See the following diagram for details.	

PIP Position diagram



PIP Scrolling Duagram



12.6 Operation during Wait/Halt/Standby & Wakeup Procedure

In wait mode, the TFT is able to remain functioning even CPU is turned off because GPL162002A keeps PLL clock activating. On the other hand, the PLL clock is shutdown in both halt and standby modes and it will cause TFT unable to display in such modes. Therefore, Generalplus recommends setting the TFTEN control bit to "0" before entering halt mode and standby mode. After GPL162002A wakes up from those modes, set the TFTEN control bit to "1" again. Simply, programmer is able to turn on/off LCD panel by configuring TFTEN control bit. The GPL162002A will automatically perform the TFT power on/off procedures in order to avoid unexpected lines occurring on the LCD panel.

12.7 Programming Example

320 (Segment) x 240 (Common) color, 8-bit interface, one PIP

```
.DEFINE          SEGMENT          320
.DEFINE          COMMON          240

r1=0x0800
[P_TFT_Ctrl]=r1
r1=0x01          //TFT clock = system clock/2
[P_TFT_DCLK_Ctrl]=r1
r1=1715          //TFT Hsync timing setup
[P_TFT_H_Width]=r1
r1=239
[P_TFT_H_Start]=r1
r1=1199
[P_TFT_H_End]=r1
r1=0x0000
[P_TFT_HSync_Setup]=r1
r1=261          //TFT Vsync timing setup
[P_TFT_V_Width]=r1
r1=20
[P_TFT_V_Start]=r1
r1=260
[P_TFT_V_End]=r1
r1=0x0000
[P_TFT_VSync_Setup]=r1
r1=0x8003          //set serial mode, set odd, even RGB data
[P_TFT_RGB_Ctrl]=r1          //type
r1=T_BMP
r1+=2          //get pic address
r2=[r1++]
[P_TFT_DMASStart_AL]=r2
```



```

r2=[r1]
[P_TFT_DMASStart_AH]=r2
r1= SEGMENT -1           //set X pixel
[P_TFT_Pixel_Num]=r1
r1= COMMON-1             //set Y pixel
[P_TFT_Line_Num]=r1

////----- //PIP setup
r1=T_DIP                 //get PIP pic address
r2=[r1++]
[P_TFT_PIP0_STR_AL]=r2
r1=[r1]
[P_TFT_PIP0_STR_AH]=r1
r1=PIP0_H_Start          // set PIP X-axis position
[P_TFT_PIP0_H_Start]=r1
r1=PIP0_H_End
[P_TFT_PIP0_H_End]=r1
r1=PIP0_V_Start          //set PIP Y-axis position
[P_TFT_PIP0_V_Start]=r1
r1=PIP0_V_End
[P_TFT_PIP0_V_End]=r1    //enable PIP controller
r1=0x8000
[P_TFT_PIP0_Ctrl]=r1

r1=[P_TFT_Ctrl]          //enable TFT controller
r1|=0x8000
[P_TFT_Ctrl]=r1

```

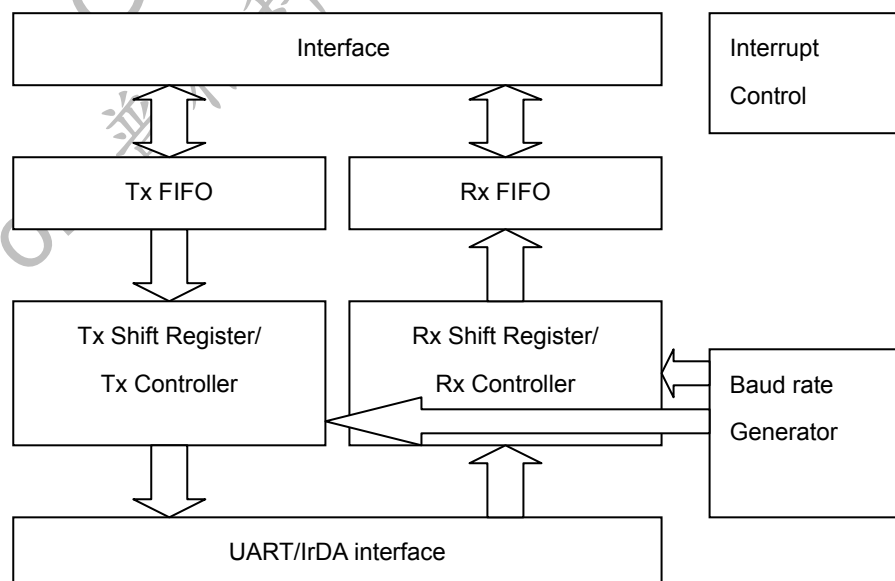
13 UART/IrDA Interface

13.1 Introduction

The UART/ IrDA module built in GPL162002A/162003A performs serial-to-parallel conversion on data received from an external device and it also performs parallel-to-serial conversion on data transmitted to the external device. The transmission and reception paths are individually buffered with internal 8 bytes FIFO memories. This module provides the following features.

- Programmable using of UART or IrDA SIR input/output.
- Data width can be 5, 6, 7, or 8 bits.
- Parity can be even, odd or disabled for generation and detection.
- Stop bit width can be 1 or 2 bits.
- Separate 8-byte transmitting and 8-byte receiving FIFOs.
- Programmable baud rate generator.
- Independent masking of transmitting FIFO, receiving FIFO, and receive timeout interrupts.
- False start bit detection.
- Link break generation and detection.
- Support normal 3/16 and low power (1.63us) bit duration.
- Programmable IrDA TX and RX latency.
- Programmable IrDA TX and RX signal polarity.
- Support Loop Back Testing for system diagnosis or mass production testing.

13.2 Structure and Block Diagram

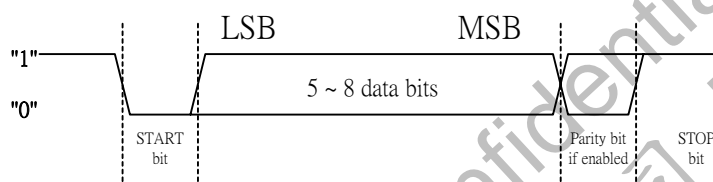


GPL162002A/162003A contains a module that manipulates both UART and IrDA signals. It is impossible to operate transmitting and receiving functions with UART and IrDA at the same time.

IrDA interface, compatible with SIR (Serial Infrared) level IrDA, accompanied with UART Interface is built in GPL162002A/162003A. This interface and an external IrDA transceiver module, connected via PortC9 and PortC10, implement IrDA physical layer. In addition, this built-in half-duplex IrDA interface also provides programmable latency and programmable signal pulse duration.

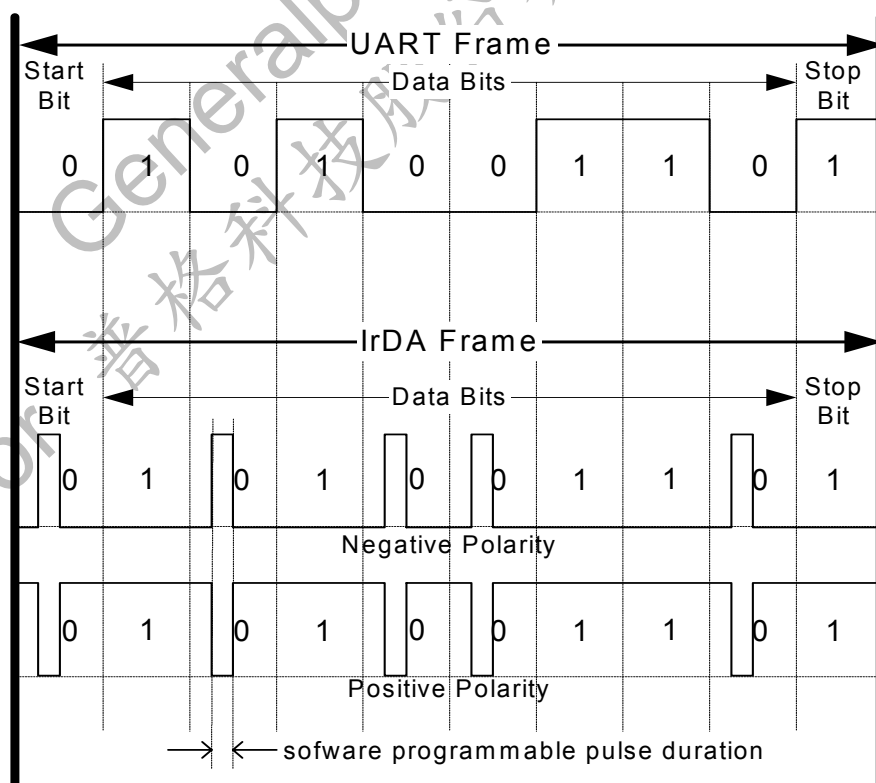
13.3 UART / IrDA SIR Frame Scheme

UART characteristic frame is depicted in the following diagram.



Note that "1" and "0" mean logical level respectively and also represent VCC and GND in GPL162002A/162003A; on the other hand, "1" and "0" mean -12V and +12V in RS232 compatible line.

IrDA SIR characteristic frame is depicted in the following diagram.



According to IrDA physical layer specification, the upper and lower limits of pulse width duration vary in various signal rates. The following table is the relative specification for IrDA SIR physical layer. Programmers must ensure to meet the IrDA SIR specification.

Signal rate	Modulation	Rate tolerance (% of rate)	Pulse duration (Minimum)	Pulse duration (Normal)	Pulse duration (Maximum)
2400 bit/s	RZI	±0.87	1.41 us	78.13 us	88.55 us
9600 bit/s	RZI	±0.87	1.41 us	19.53 us	22.13 us
19200 bit/s	RZI	±0.87	1.41 us	9.77 us	11.07 us
38400 bit/s	RZI	±0.87	1.41 us	4.88 us	5.96 us
57600 bit/s	RZI	±0.87	1.41 us	3.26 us	4.34 us
115200 bit/s	RZI	±0.87	1.41 us	1.63 us	2.23 us

13.4 UART/IrDA Control Pin Configuration

Name	I/O	Description
URX	I	UART Reception Pin (shared with PortC10)
UTX	O	UART Transmission Pin (shared with PortC9)
IRRX	I	IrDA SIR Reception Pin (shared with PortC10)
IRTX	O	IrDA SIR Transmission Pin (shared with PortC9)

13.5 Control registers

UART/IrDA Control Register Summary Table

Name	Address	Description
P_UARTIrDA_Data	0x7900	UART/ IrDA Data Register
P_UART_RXStatus	0x7901	UART Reception Status (Error Flag) Register
P_UARTIrDA_Ctrl	0x7902	UART/ IrDA Control Register
P_UART_BaudRate	0x7903	UART Baud Rate Setup Register
P_UARTIrDA_Status	0x7904	UART/ IrDA Interrupt Register
P_UARTIrDA_FIFO	0x7905	UART/IrDA FIFO Control Register
P_UART_TXDLY	0x7906	UART TX Delay Control Register
P_IrDA_BaudRate	0x7907	IrDA Baud Rate Setup Register
P_IrDA_Ctrl	0x7908	IrDA Control Register
P_IrDALP	0x7909	IrDA Low Power Control Register

P_UARTIrDA_Data				0x7900				UART / IrDA Data Register												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Function	-	-	-	-	-	-	-	-	UARTDATA											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	UARTDATA	R/W	UART/IrDA Data Read/Write Register	

P_UART_RXStatus 0x7901
UART Reception Error Flag Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	OE	BE	PE	FE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	OE	R/W	Overrun Error This bit is set to "1" if data is received and the FIFO is full.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
2	BE	R/W	Break Error This bit is set to "1" if a break condition is detected. Indicate that the received data input is held LOW for more than a full-word transmission time (defined as start, data, parity and stop bits). This bit is refreshing in every read. So, it is necessary to check this bit after DATA register is read.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
1	PE	R/W	Parity Error This bit is set to "1" if the parity of the received data does not match the parity selected in PSEL Control bit. This bit is refreshing in every read. So, it is necessary to check this bit after DATA register is read.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag
0	FE	R/W	Frame Error This bit is set to "1" if a received character does not have a valid stop bit (a valid stop bit is 1 bit). This bit is refreshing in every read. So, it is necessary to check this bit after DATA register is read.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear this Error Flag

Receiving status is read from **P_UART_RXStatus** control register. The status information corresponds to the data read from P_UARTIrDA_Data control register prior to reading **P_UART_RXStatus** control

register. Write “1” to corresponding register bit will clear the frame, parity, break, and overrun error.

Note that, the received data byte must be read first from P_UARTIrDA_Data before reading the corresponding error status from **P_UART_RXStatus**. This read sequence cannot be reversed since the status register **P_UART_RXStatus** is updated only when a read operation is performed on P_UARTIrDA_Data control register.

P_UARTIrDA_Ctrl
0x7902
UART / IrDA Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RXIE	TXIE	RTIE	UEN	MSIE	SLT	-	-	-	WLSEL	FEN	SBSEL	PSEL	PEN	SB	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RXIE	R/W	<p>Receive Interrupt Enable</p> <p>For FIFO is enabled (8 depth, FEN=1): If this bit is set to “1”, and the data number in receiving FIFO is more or equal to 8, hardware will issue an IRQ3 or FIQ to CPU.</p> <p>For FIFO is disable (1 depth, FEN=0): If this bit is set to “1”, and the receiving buffer is just held one new-coming data, hardware will issue an IRQ3 or FIQ to CPU.</p> <p>If this bit is cleared to “0”, this receiving interrupt will be masked. To select between IRQ3 and FIQ, please refer to Chapter Interrupt.</p>	<p>0= Disabled</p> <p>1= Enabled</p>
14	TXIE	R/W	<p>Transmit Interrupt Enable</p> <p>For FIFO is enabled (8 depth, FEN=1): If this bit is set to “1”, and the data number in transmitting FIFO is less or equal to 1, hardware will issue an IRQ3 or FIQ to CPU.</p> <p>For FIFO is disable (1 depth, FEN=0): If this bit is set to “1”, and the transmitting buffer is empty, hardware will issue an IRQ3 or FIQ to CPU.</p> <p>If this bit is cleared to “0”, this transmitting interrupt will be masked. To select between IRQ3 and FIQ, please refer to Chapter Interrupt.</p>	<p>0= Disabled</p> <p>1= Enabled</p>
13	RTIE	R/W	<p>Receive Timeout Interrupt Enable</p> <p>If this bit is set to “1”, and when the receiving FIFO is not empty and no further data is</p>	<p>0= Disabled</p> <p>1= Enabled</p>

Bit	Function	Type	Description	Condition
			received over a 32-bit period, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this receiving interrupt will be masked. To select between IRQ3 and FIQ, please refer to Chapter Interrupt .	
12	UEN	R/W	UART Enable If this bit is set to "1", the UART Interface is enabled. Data transmission and reception occur as either UART signals (UTX, URX [PortD5, PortD6]), or IrDA SIR signals (IRTX, IRRX [PortD5, PortD7]) based on the setting of Control bit IEN (P_IrDA_Ctrl.bit9). For example, to enable IrDA control module, both UEN and IEN bits should be set to "1", to enable UART control module, UEN should be set to "1", but IEN should clear to "0".	0= Disabled 1= Enabled
11	MSIE		Modem Status Interrupt Enable If this bit is set to "1", the modem status interrupt is enabled.	0= Disable 1= Enable
10	SLT	R/W	Self-Loop Test Enable This bit is used for IrDA mode. If this bit is set to "1", IrDA device is seems as full-duplex.	0= Disabled 1= Enabled
[9:7]			Reserved	
[6:5]	WLSEL	R/W	Word Length Definition Indicate number of data bits transmitted or received in a frame.	00= 5 bits 01= 6 bits 10= 7 bits 11= 8 bits
4	FEN	R/W	FIFO Buffer Enable/Disable Setting this bit to "1" will enable 16-depth FIFO buffer on receiving operation and 2-depth FIFO buffer on transmitting operation. When clearing this bit to "0", the FIFO becomes 1-byte-deep hold registers.	0= Disabled 1= Enabled
3	SBSEL	R/W	Stop Bit Size Selection When this bit is set to "1", two stop bits are transmitted at the end of the frame. The receiving logic cannot check for data with two received stop bits.	0= 1 Stop Bit 1= 2 Stop Bit
2	PSEL	R/W	Parity Selection If this bit is set to "1", even parity generation and checking are performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to "0"	0= Odd Parity (if PEN= 1) 1= Even Parity (if PEN= 1)

Bit	Function	Type	Description	Condition
			the odd parity is performed to check for an odd number of 1s. This bit has no effect until parity is enabled by setting PEN Control bit to "1".	
1	PEN	R/W	Parity Enable If this bit is set to "1", parity checking and generation is enabled, or else parity is disabled and no parity bit is added to the data frame.	0= Disabled 1= Enabled
0	SB	R/W	Send Break If this bit is set to "1", a low level is continuously output on the TX output pin after completing the current character transmission. This bit must be asserted for at least one complete frame transmission time in order to generate a break condition. The transmitting FIFO contents remain ineffective during a break condition. For normal usage, this bit must be cleared to "0".	0= Normal Operation 1= Send Break Signal

P_UART BaudRate 0x7903
UART Baud Rate Setup Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BUAD															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BUAD	R/W	UART Baud Rate control	

The Baud rate = system clock / BUAD

For example: system clock is 48MHz and 115200 bps of UART baud rate is desired.

BUAD= 48000000/115200

P_UARTIrDA_Status 0x7904
UART / IrDA Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RXIF	TXIF	RTIF	-	-	-	-	-	TXEF	RXFF	TXFF	RXEF	BY	DCD	DSR	CTS
Default	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
15	RXIF*	R	Receive Interrupt Flag For FIFO is enabled (8 depth, FEN=1): This bit is set to "1" by hardware if the receiving interrupt enable bit is set to "1", and the data number in receiving	If FIFO is enabled: Read 0= no. in RX FIFO < 8 Read 1= no. in RX FIFO >= 8 If FIFO is disabled: Read 0= RX Data is not Ready Read 1= RX Data is Ready

Bit	Function	Type	Description	Condition
			<p>FIFO is more or equal to eight. This bit is cleared to “0” by hardware if data number in receiving FIFO is less than eight. In other word, receive interrupt is cleared by reading data from the receive FIFO until it becomes less than eight data.</p> <p>For FIFO is disabled (1 depth, FEN=0):</p> <p>This bit is set to “1” by hardware if the receiving interrupt enable bit is set to “1”, and the data is received completely. The receive interrupt is cleared by performing a single read of receive data register (P_UARTIrDA_Data).</p>	
14	TXIF*	R	<p>Transmit Interrupt Flag</p> <p>For FIFO is enabled (8 depth, FEN=1):</p> <p>This bit is set to “1” by hardware if the transmitting interrupt enable bit is set to “1”, and the data number in transmitting FIFO is less or equal to one. This bit is cleared to “0” by hardware if data number in transmitting FIFO is more than one. In other word, transmitting interrupt is cleared by filling data to transmitting FIFO until it becomes more than one data.</p> <p>For FIFO is disabled (1 depth, FEN=0):</p> <p>This bit is set to “1” by hardware if the transmitting interrupt enable bit is set to “1”, and there is no data in the transmitter. This flag is cleared to “0” by performing a single write to the transmit data register (P_UARTIrDA_Data).</p>	<p>If FIFO is enabled:</p> <p>Read 0= no. in TX FIFO > 8 Read 1= no. in TX FIFO ≤ 8</p> <p>If FIFO is disabled:</p> <p>Read 0= TX Buffer is not Ready Read 1= TX Buffer is Ready</p>
13	RTIF*	R	<p>Receive Timeout Interrupt Flag</p> <p>This bit is set to “1” by hardware if the receiving timeout interrupt is asserted. (When receiving FIFO is not empty and no further data is received over a 32-bit</p>	<p>Read 0= Not Occurred Read 1= Occurred</p>

Bit	Function	Type	Description	Condition
			period). This interrupt flag is cleared to “0” by hardware when the FIFO becomes empty by reading all data or by reading the holding register	
[12:8]			Reserved	
7	TXEF	R	Transmit FIFO Empty Flag The meaning of this bit depends on the state of the FEN control bit. For FIFO is enabled (2 depth, FEN=1): This bit is set to “1” when the transmitting FIFO is empty. For FIFO is disabled (1 depth, FEN=0): This bit is set to “1” when the transmitting hold register is empty. Note that this flag is read-only; hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in TX FIFO > 0 Read 1= no. in TX FIFO = 0 If FIFO is disabled: Read 0= TX buffer is not empty Read 1= TX buffer is empty
6	RXFF	R	Receive FIFO Full Flag The meaning of this bit depends on the state of the FEN control bit. For FIFO is enabled (8 depth, FEN=1): This bit is set to “1” when the receive FIFO is full. For FIFO is disabled (1 depth, FEN=0): This bit is set to “1” when the receiving hold register is full. Note that this flag is read-only; hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in RX FIFO < 8 Read 1= no. in RX FIFO = 8 If FIFO is disabled: Read 0= RX buffer is not full Read 1= RX buffer is full
5	TXFF	R	Transmit FIFO Full Flag The meaning of this bit depends on the state of the FEN control bit. For FIFO is enabled (8 depth, FEN=1): This bit is set to “1” when the transmitting FIFO is full. For FIFO is disabled (1 depth, FEN=0): This bit is set to “1” when the transmitting hold register is full.	If FIFO is enabled: Read 0= no. in TX FIFO < 8 Read 1= no. in TX FIFO = 8 If FIFO is disabled: Read 0= TX buffer is not full Read 1= TX buffer is full

Bit	Function	Type	Description	Condition
			Note that this flag is read-only; hardware will set or clear this flag automatically.	
4	RXEF	R	Receive FIFO Empty Flag The meaning of this bit depends on the state of the FEN control bit. For FIFO is enabled (8 depth, FEN=1): This bit is set to "1" when the receiving FIFO is empty. For FIFO is disabled (1 depth, FEN=0): This bit is set to "1" when the receiving hold register is empty. Note that this flag is read-only; hardware will set or clear this flag automatically.	If FIFO is enabled: Read 0= no. in RX FIFO > 0 Read 1= no. in RX FIFO = 0 If FIFO is disabled: Read 0= RX buffer is not empty Read 1= RX buffer is empty
3	BY	R	BUSY When this bit is read as "1", the UART or IrDA module is busy in transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. Note that this flag is read-only; hardware will set or clear this flag automatically.	0= Ready 1= Busy
2	DCD	R	This bit is the complement of the nUARTDCD modem status input.	0= nUARTDCD is 1 1= nUARTDCD is 0
1	DSR	R	This bit is the complement of the nUARTDSR modem status input.	0= nUARTDSR is 1 1= nUARTDSR is 0
0	CTS	R	This bit is the complement of the nUARTCTS modem status input.	0= nUARTCTS is 1 1= nUARTCTS is 0

*The above three interrupt flag are combined into a single interrupt flag on P_INT_Status1.bit11 by an OR logic. Programmers should determine which interrupt occurs from this control register (P_UARTIrDA_Status.bit [15...13]).

P_UARTIrDA_FIFO 0x7905
UART/IrDA FIFO Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	TX_LEVEL			-	TX_FLAG			-	RX_LEVEL			-	RX_FLAG		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15			Reserved	
[14:12]	TX_LEVEL	R/W	Transmit FIFO interrupt level register This register is used to indicate how many empty slots are required when issuing an interrupt. The larger the value is set, the lower the interrupt penalty you have since you can write more data in one interrupt.	FIFO Full Interrupt issue timing. 000 = data no. in FIFO < 1, 8 write is allowed. 001 = data no. in FIFO < 2, 7 write is allowed. 010 = data no. in FIFO < 3, 6 write is allowed. 011 = data no. in FIFO < 4, 5 write is allowed. 100 = data no. in FIFO < 5, 4 write is allowed. 101 = data no. in FIFO < 6, 3 write is allowed. 110 = data no. in FIFO < 7, 2 write is allowed. 111 = data no. in FIFO < 8, 1 write is allowed.
11			Reserved	
[10:8]	TX_FLAG	R	Transmit FIFO Data Level This register indicates how many data remain in transmit FIFO.	000 = 0 byte in FIFO. 001 = 1 byte in FIFO. 010 = 2 bytes in FIFO. 011 = 3 bytes in FIFO. 100 = 4 bytes in FIFO. 101 = 5 bytes in FIFO. 110 = 6 bytes in FIFO. 111 = 7 bytes in FIFO.
7			Reserved	
[6:4]	RX_LEVEL	R/W	This register is used to indicate how many bytes are stored in receiving FIFO when issuing an interrupt. The larger the value is set, the lower the interrupt penalty you have since you can read more data in one interrupt.	FIFO Full Interrupt issue timing. 000 = data no. in FIFO >= 1, 1 read is allowed. 001 = data no. in FIFO >= 2, 2 read is allowed. 010 = data no. in FIFO >= 3, 3 read is allowed. 011 = data no. in FIFO >= 4, 4 read is allowed. 100 = data no. in FIFO >= 5, 5 read is allowed. 101 = data no. in FIFO >= 6, 6 read is allowed. 110 = data no. in FIFO >= 7, 7 read is allowed.

Bit	Function	Type	Description	Condition
				111= data no. in FIFO >= 8, 8 read is allowed.
3			Reserved	
[2:0]	RX_FLAG	R	Receive FIFO Data Level This register indicates how many data have been received in receive FIFO.	000 = 0 byte in FIFO 001 = 1 byte in FIFO 010 = 2 bytes in FIFO 011 = 3 bytes in FIFO 100 = 4 bytes in FIFO 101 = 5 bytes in FIFO 110 = 6 bytes in FIFO 111 = 7 bytes in FIFO

P_UART_TXDLY
0x7906
UART TX Delay Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	TWT	R/W	Transmitter Waiting time. It is used to make a delay between two transmitting bits.	0 = no delay 0001~1111 = 1~15 bits delay

P_IrDA_BaudRate
0x7907
IrDA Baud Rate Setup Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BUAD															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BUAD	R/W	IrDA Buad Rate control	

The Buad rate = system clock / BUAD

For example: system clock is 48MHz and 9600bps of UART buad rate is desired. Then

BUAD= 48000000/9600

P_IrDA_Ctrl
0x7908
IRDA Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TXLT				TPOL	RPOL	IEN	ILP	RXLT							
Default	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]	TXLT	R/W	Transmit Latency Time It defines the delay time between the time that the transmitter is finished and the time that the receiver starts.	0= no delay 1~15= 1~15 bit delay time
11	TPOL	R/W	IrDA Transmission Polarity Selection	1: Negative Polarity 0: Positive Polarity
10	RPOL	R/W	IrDA Reception Polarity Selection	1: Positive Polarity 0: Negative Polarity
9	IEN	R/W	IrDA SIR Enable If this bit is set to "1", the IrDA SIR Endec is enabled. This bit has no effect if the UART is not enabled by setting UEN Control bit as "1". When the IrDA SIR Endec is enabled, IrDA data is transmitted and received on PortC9 and PortC10 respectively. In other words, these two pins cannot be used as GPIO at this time. When the IrDA SIR Endec is disabled, PortC9 and PortC10 are able to function as GPIO.	0= Disabled 1= Enabled
8	ILP	R/W	IrDA SIR Low Power Mode Selection If this bit is cleared to "0", low level data bits are transmitted as an active high pulse with 3/16 th of a bit period. If this bit is set to "1", low level data bits are transmitted with a pulse width which is 3 times of the period of an input signal. Note that setting this bit consumes less power, but the transmission distance may become shorter.	0= normal mode, fixed to 3/16 th of corresponding bit period 1= low power mode, fixed to 1.63us
[7:0]	IrDAPD	R/W	IrDA SIR Receive Latency Setup It defines the delay time between the time that the receiver is finished and the time that the transmitter starts.	0= no delay 1~255= 1~255 bits delay time

P_IrDALP
0x7909
IrDA Low Power Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	IrDALP							
Default	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	IrDALP	R/W	IrDA Low Power Divisor	

13.6 Program Examples

This sample code is self-loop test of UART module. To make it work, IOC9 and IOC10 should be interconnected.

```

        r1 = 48000000/115200                // For System clock = 48MHz
        [P_UARTIrDA_BaudRate] = r1         // Baud Rate = 115,200 bps
        r1 = 0x9070
        [P_UARTIrDA_Ctrl] = r1             //UART Enable, N-8-1, FIFO Enable
        r4 = 0x0050
        [P_UARTIrDA_Data] = r4
        r4 += 1                            //1st TX FIFO data = 0x50
        r1 = 7

L_FillRXFIFO?:
        [P_UARTIrDA_Data] = r4             //2nd, 3rd, 4th, 5th, 6th, 7th, 8th TX FIFO data
        r1 = r1 - 1                        // = 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57
        r4 = r1 + 1
        jnz L_FillRXFIFO?

L_WaitRX_FIFO_Empty_INT?:
        r1 = [P_UARTIrDA_Status]           // If data number of RX FIFO is more than or
        jpl L_WaitRX_FIFO_Empty_INT?       // equal to 8, RX interrupt flag will be set to "1"

        r4 = 0x50
L_Wait_RX_FIFO_Empty?:
        r1 = [P_UARTIrDA_Status]           // Verify all RX FIFO data
        r1 = r1 & 0x0010
        jnz L_UARTSelfLoopTestOK?
        r3 = [P_UARTIrDA_Data]
        cmp r3,r4
        jne L_UARTSelfLoopTestError?
        r4 = r4 + 1
        Jmp L_Wait_RX_FIFO_Empty?

L_UARTSelfLoopTestOK?:
        .....

L_UARTSelfLoopTestError?:
        .....

```

14 Serial Peripheral Interface (SPI)

14.1 Introduction

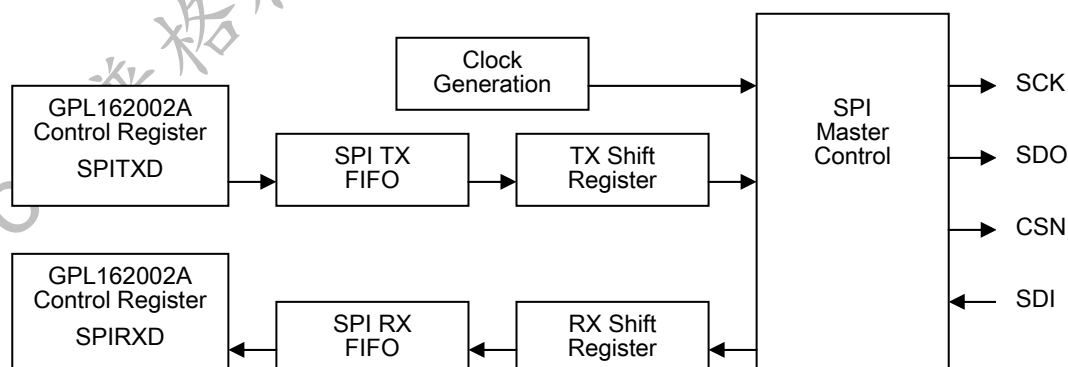
A Serial Peripheral Interface (SPI) controller is built in GPL162002A/162003A to facilitate communicating with other devices and components. The built-in SPI controller GPL162002A/162003A includes four master mode and one slave mode. There are four control signals on SPI, SPICSN, SPICLK (SCK), SPIDI (SDI), SPIDO (SDO); these signals are shared with PortB10, Portb11, PortD11 and PortD4 and SPICSN is valid only on slave mode. While SPI module is enabled by setting corresponding control bit, these four pins cannot be GPIOs. In other words, any setting on corresponding GPIO control register will have no effect.

The SPI provides following features:

- Selectable single-byte or consecutive-byte transfers.
- Overrun error indication.
- Transmitting / receiving interrupt requests.
- Programmable phase and polarity of master clock.
- Selectable data sampling time (at the end or middle of clock period).
- Programmable master SCK clock frequency: System Clock /2, / 4, /8, /16, /32, /64, /128.
- Built-in 8-depth 8-bits FIFOs in both transmitting and receiving direction with programmable interrupt level.

14.2 Structure

Following is a function diagram of SPI module



14.3 SPI Control Pin Configuration

Name	I/O	Description
SPICK	O	Serial Peripheral Interface, Clock Pin (Shared with PortB11)
SPICSN	O	Serial Peripheral Interface, Chip Select, Low Active (Shared with PortB10)
SPIDO	O	Serial Peripheral Interface, Data Out Pin (Shared with PortD4)
SPIDI	I	Serial Peripheral Interface, Data In Pin (Shared with PortD11)

14.4 Master Mode

In master mode, the shifting clock (SPICLK) is generated by GPL162002A/162003A. There are two control bits to control the clock phase and polarity. The transmission starts immediately from writing SPIBUF control register. As long as there is a data in the FIFO, the transmission will start automatically.

The SPI shifts the data from MSB to LSB through the SDO pin. The 8-bit data is shifted out after 8 SCK cycles. At the same time, the data is also shifted in through slave device SDI pin. When the transmitting FIFO level is lower than the interrupt trigger level, the SPITXIF flag bit will be set; besides, a SPI interrupt will be generated if the SPITXIRQEN bit is set. When the receiving FIFO level is higher than the interrupt trigger level, the SPIRXIF flag bit will be set; besides, a SPI interrupt will be generated if the SPIRXIRQEN bit is set. Programmers can read SPI data from SPIRXD control register.

The following diagram depicts the timing scheme on SPI master mode for different operation types (polarity control bit equals "1" or "0", phase control bit equals "1" or "0", and sample strobe control bit equals "1" or "0").

14.5 Slave Mode

In slave mode, the SPICLK becomes an input pin that receives external clock. And all clock and data are valid when SPICSN is at low state. In GPL162002A/162003A SPI slave mode only supports polarity=0 and phase=0 mode.

Each time when starting to transmit data to a slave GPL162002A/162003A, the SPICSN needs to switch to low for GPL162002A/162003A SPI clock and data synchronous and needs to switch SPICSN to high when data transmission is finished.

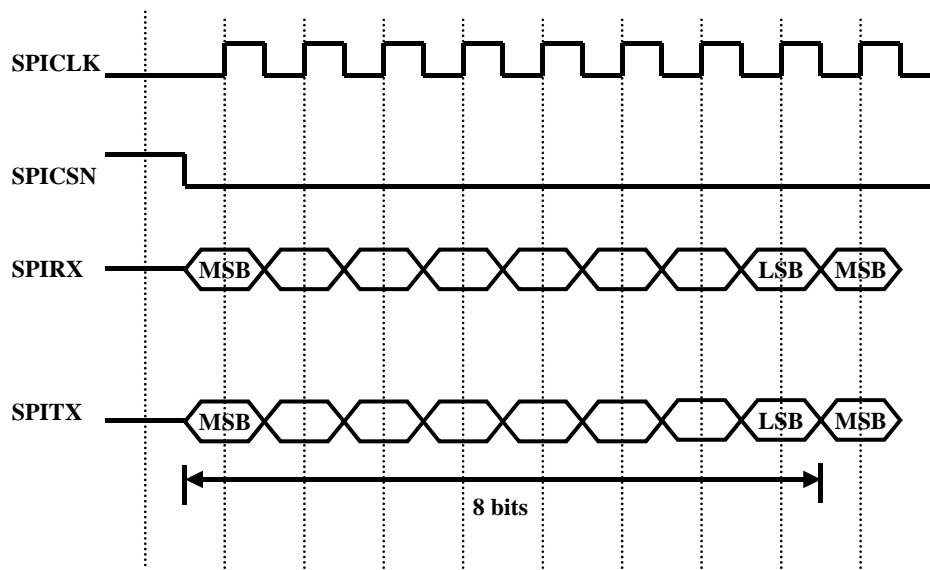


Fig1. Master Mode, POLARITY=0, PHASE=0

At this setting, GPL162002A/162003A will sample data at rising edge of SPICLK.

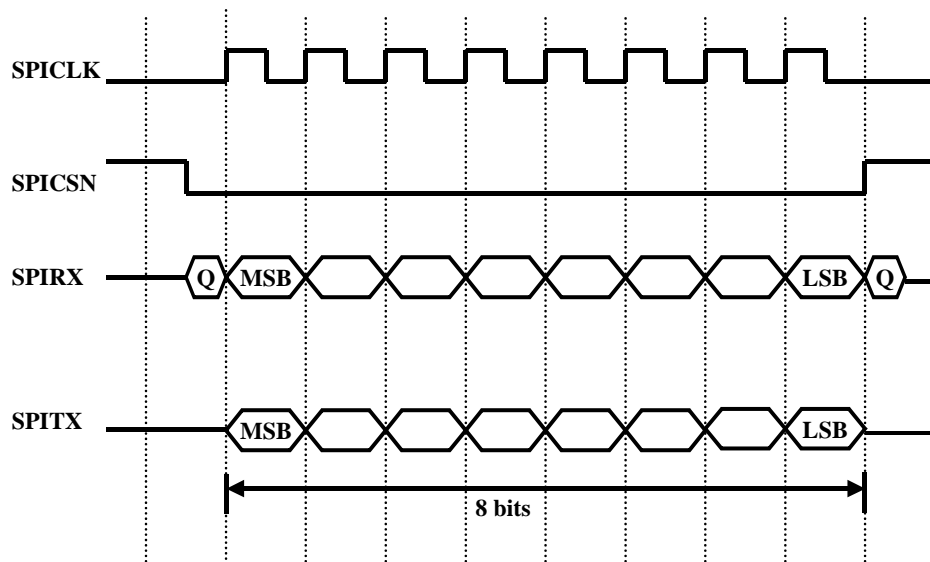


Fig2. Master Mode, POLARITY=0, PHASE=1

At this setting, GPL162002A/162003A will sample data at falling edge of SPICLK.

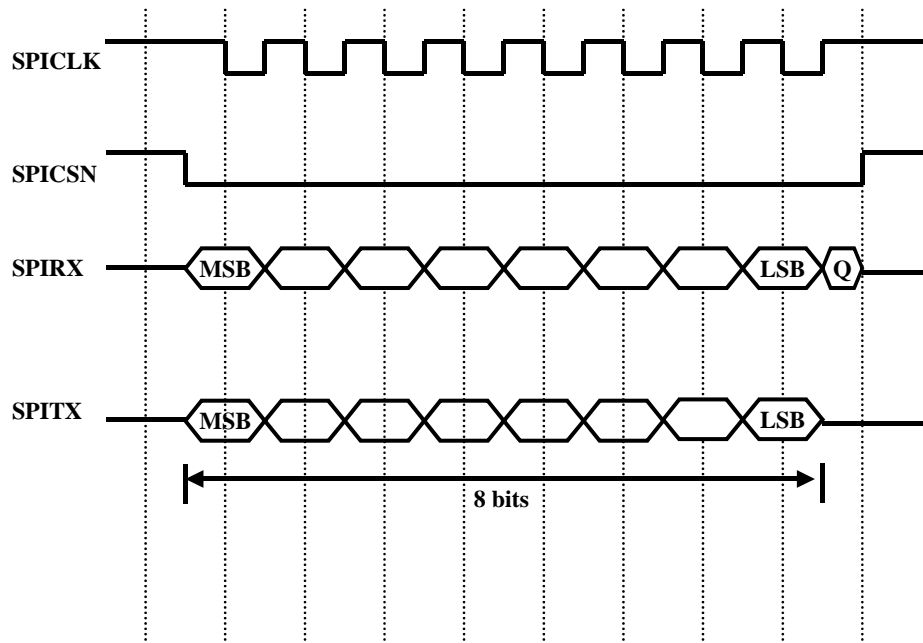


Fig3. Master Mode, POLARITY=1, PHASE=0

At this setting, GPL162002A/162003A will sample data at falling edge of SPICLK.

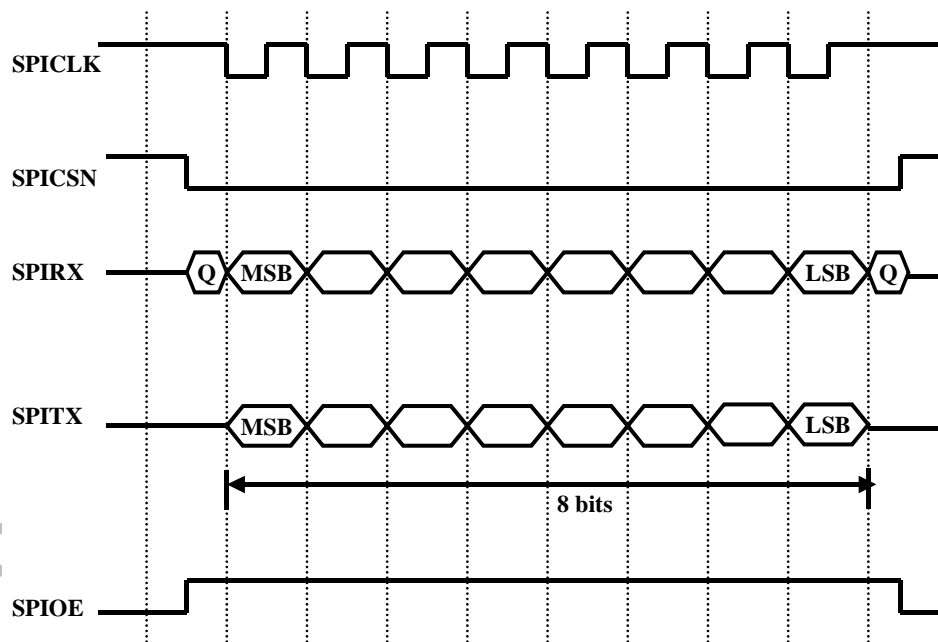


Fig4. Master Mode, POLARITY=1, PHASE=1

At this setting, GPL162002A/162003A will sample data at rising edge of SPICLK.

14.6 Consecutive Bytes Transfer

Consecutive-byte transfers are available in the master mode. In transmission, software is able to send the data consecutively as long as the SPITXBF flag is not set. In reception, software should check for overrun error to monitor if there is any missing data in case the polling rate is too low.

14.7 Control Registers

SPI Control Register Summary Table

Name	Address	Description
P_SPI_Ctrl	0x7940	SPI Control Register
P_SPI_TXStatus	0x7941	SPI Transmit Status Register
P_SPI_TXData	0x7942	SPI Transmit FIFO Register
P_SPI_RXStatus	0x7943	SPI Receive Status Register
P_SPI_RXData	0x7944	SPI Receive FIFO Register
P_SPI_Misc	0x7945	SPI Misc Control Register

P_SPI_Ctrl		0x7940								SPI Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	SPIEN	-	LBM	-	SPIRST	-	-	MOD	-	-	SCKPHA	SCKPOL	-	SCKSEL			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	SPIEN	R/W	SPI enable If this bit is set to "1", and MOD=0, PortB [11:10] and PortD4, PortD11 become a SPI Interface. If this bit is set to "1", and MOD=1, PortB11 and PortD4, PortD11 become a SPI Interface. These pins cannot be used as GPIO once SPI is enabled. Therefore, any further setting on the selected GPIOs will be no effect.	0= Disabled 1= Enabled
			Reserved	
13	LBM		Loop Back Mode Selection When this bit is set to 1, the SPIRX will be connected to SPITX internally. It is for test only.	0 = Normal Mode 1 = SPIRX = SPITX
12			Reserved	
11	SPIRST	W	SPI Soft Reset If this bit is written by "1", the state machine of SPI controller and FIFO pointer will return to the original value.	0= No effect 1= Reset SPI Controller
[10:9]			Reserved	

Bit	Function	Type	Description	Condition
8	MOD	R/W	SPI Mode Selection register In slave mode, GPL162002A/162003A only supports SCKPHA=0, SCKPOL=0 timing. And SPICSN becomes SPI chip select pin. In master mode, GPL162002A/162003A supports four SPI timing settings. And SPICSN does not become SPI interface.	0= Master 1= Slave
[7:6]			Reserved	
5	SCKPHA	R/W	SPI clock phase This bit should not be changed after SPIEN is 1.	Refer to timing scheme on previous section
4	SCKPOL	R/W	SPI clock polarity This bit should not be changed after SPIEN is 1.	Refer to timing scheme on previous section
3			Reserved	
[2:0]	SCKSEL	R/W	Master mode clock selection	000= SYSCLK / 2 001= SYSCLK / 4 010= SYSCLK / 8 011= SYSCLK / 16 100= SYSCLK / 32 101= SYSCLK / 64 110 = SYSCLK / 128 111 = Reserved

P_SPI_TXStatus
0x7941
SPI Transmit Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SPITXIF	SPITXIEN	-	-	-	-	-	-	-	TXFLEV			TXFFLAG			
Default	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	SPITXIF	R/W	SPI Transmit Interrupt flag This bit is set to "1" by hardware when the transmitting FIFO level is lower then the value setting by users. When SMART is set in P_SPI_Misc register, the bit will be cleared as long as the transmitting FIFO level is higher than interrupt level; else users should write "1" to clear this flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	SPITXIEN	R/W	SPI Transmit Interrupt Enable If this bit is set to "1" and SPI interrupt (when 8-bit TX FIFO level in lower then interrupt level) occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0 = Disabled 1 = Enabled Note: If enable SPI TX interrupt, Generalplus suggest programmer enable smart interrupt clear function. (P_SPI_Misc .b8)

Bit	Function	Type	Description	Condition
			To select between IRQ3 and FIQ, please refer to Chapter Interrupt .	
[13:8]			Reserved.	
[7:4]	TXFLEV	R/W	Transmit FIFO interrupt level register. This register is used to indicate how many empty slots are required when issuing an interrupt. The larger the value is set, the lower the interrupt penalty users have since users can write more data in one interrupt.	FIFO Full Interrupt issue timing 0000 = data no. in FIFO < 1, 8 write is allowed. 0001 = data no. in FIFO < 7, 2 write is allowed. 0010 = data no. in FIFO < 6, 3 write is allowed. 0011 = data no. in FIFO < 5, 4 write is allowed. 0100 = data no. in FIFO < 4, 5 write is allowed. 0101 = data no. in FIFO < 3, 6 write is allowed. 0110 = data no. in FIFO < 2, 7 write is allowed. 0111 = data no. in FIFO < 1, 8 write is allowed. 1000~1111 = not valid
[3:0]	TXFFLAG	R	Transmit FIFO Data Level The register is used to indicate how many data are still in the FIFO.	0000 = No data in FIFO or 8 bytes in FIFO. 0001 = 1 byte in FIFO. 0010 = 2 bytes in FIFO. 0011 = 3 bytes in FIFO. 0100 = 4 bytes in FIFO. 0101 = 5 bytes in FIFO. 0110 = 6 bytes in FIFO. 0111 = 7 bytes in FIFO.

P SPI TXData
0x7942
SPI Transmit FIFO Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	SPIDATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	SPITXDATA	W	Write data to SPI Transmit FIFO.	

P_SPI_RXStatus							0x7943							SPI Transmit Status Register						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Function	SPITXIF	SPITXIEN	-	-	-	-	RXFULL	RXFOV	TXFLEV				TXFFLAG							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
15	SPIRXIF	R/W	SPI Transmit Interrupt flag This bit is set to "1" by hardware when the receiving FIFO level is higher then the value set by users. When SMART is set in P_SPI_Misc register, the bit will be cleared as long as the receiving FIFO level is lower than interrupt level; else users should write "1" to clear this flag.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	SPIRXIEN	R/W	SPI Receive Interrupt Enable If this bit is set to "1", and SPI interrupt (when 8-bit RX FIFO level in higher then interrupt level) occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ3 and FIQ, please refer to Chapter Interrupt.	0 = Disabled 1 = Enabled Note: If enable SPI RX interrupt, Generalplus suggest programmer enable smart interrupt clear function. (P_SPI_Misc .b8)
[13:10]			Reserved.	
9	RXFULL	R	Receive FIFO full register This bit will be set by hardware when the receiving FIFO is full. In addition, this bit will be clear by hardware when the receiving FIFO is not full.	0 = Not Full 1 = Full
8	RXFOV	R/W	Receive FIFO over run register If the RX FIFO is full, and a data is coming in immediately, this flag will be set. The new coming data will overwrite the last input data or be skipped, please refer to P_SPI_Misc.	Read 0= Not occurred Read 1= Occurs Write 0= No effect Write 1= Clear the flag
[7:4]	RXFLEV	R/W	Receive FIFO interrupt level register This register is used to indicate how many bytes are stored in receiving FIFO when issuing an interrupt. The larger the value is set, the lower the interrupt penalty users have since users can read more data in one interrupt.	FIFO Full Interrupt issue timing 0000 = data no. in FIFO >= 1, 1 read is allowed. 0001 = data no. in FIFO >= 2, 2 read is allowed. 0010 = data no. in FIFO >= 3, 3 read is allowed. 0011 = data no. in FIFO >= 4, 4 read is allowed.

Bit	Function	Type	Description	Condition
				0100 = data no. in FIFO >= 5, 5 read is allowed. 0101 = data no. in FIFO >= 6, 6 read is allowed. 0110 = data no. in FIFO >= 7, 7 read is allowed. 0111 = data no. in FIFO >= 8, 8 read is allowed. 1000~1111 = not valid
[3:0]	RXFFLAG	R	Receive FIFO Data Level The register is used to indicate how many data are still in the FIFO.	0000 = No data in FIFO or 8 bytes in FIFO. 0001 = 1 byte in FIFO. 0010 = 2 bytes in FIFO. 0011 = 3 bytes in FIFO. 0100 = 4 bytes in FIFO. 0101 = 5 bytes in FIFO. 0110 = 6 bytes in FIFO. 0111 = 7 bytes in FIFO.

P_SPI_RXData
0x7944
SPI Receive FIFO Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	SPIDATA						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	SPIRXDATA	R	Read data from SPI Transmit FIFO.	

P_SPI_Misc
0x7945
SPI Misc. Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	OVER	SMART	-	-	-	BSY	RFF	RNE	TNF	TFE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Function	Type	Description	Condition
[15:10]			Reserved.	
9	OVER	R/W	SPI FIFO Over Write Mode This register is used to control the data which will be overwrite or skipped when TX/RX FIFO is full.	0 = The further write to the full FIFO will be skipped. 1 = The further write to the full FIFO will overwrite the last written data in the FIFO.
8	SMART	R/W	SPI FIFO SMART Mode Register	0 = Normal Interrupt Clear

Bit	Function	Type	Description	Condition
			When this bit is set to "1", programmers do no need to clear the transmitting/receiving interrupt flag when the FIFO status is reached, programmers only need to write to(read from) transmitting/receiving FIFO and keep the FIFO level lower/higher than the interrupt level, and the interrupt flag will be cleared automatically.	1 = Smart Interrupt Clear Note: Generalplus suggest programmer enable this bit for SPI function work correctly.
[7:5]			Reserved.	
4	BSY	R	SPI Controller Busy Flag This bit is used to indicate if the SPI controller is busy or not.	0 = Idle 1 = Busy
3	RFF	R	Receive FIFO Full Flag This bit is used to indicate if the SPI controller is real full or not. If the receive FIFO is full, any data read from SPI bus cannot be written into the FIFO, and the RFOV bit will be set in this situation.	0 = Receive FIFO not full. 1 = Receive FIFO full, not empty slot.
2	RNE	R	Receive FIFO Not Empty Flag This bit is used to indicate if there is any data currently in the receive FIFO.	0 = Receive FIFO is empty. 1 = Receive FIFO is not empty.
1	TNF	R	Transmit FIFO Not Full Flag This bit is used to indicate if there is any empty slot in the transmitting FIFO.	0 = Transmit FIFO is full; you can't write any more data into it. 1 = Transmit FIFO is not full.
0	TFE	R	Transmit FIFO Empty Flag This bit is used to indicate if the transmitting FIFO is empty or not.	0 = Transmit FIFO is not empty. 1 = Transmit FIFO is empty.

14.8 Program Examples

SPI self-loop test, the SPIDI, SPIDO need be connected.

F_SPISelfLoopTest:

Int off

```
r1=0xa100 // Enable SPI, Master mode, self-loop test
[P_SPI_Ctrl]=r1 //phase 0, polarity 0
```

```
r1 = 0x0050 // Output 0x50 to SPI
[P_SPI_TXData] = r1
```

RX_Ready?:

```
r2=[P_SPI_RXStatus]
test r2,0x01
jz RX_Ready?
```

```
r2=[P_SPI_RXData]
cmp r1,r2
jne L_Error?
DisplayResultCode D_OK
retf
```

L_Error?:

```
DisplayResultCode D_NG
retf
```

//Note that **DisplayResultCode** is MACRO

15 USB Interface

15.1 Introduction

The built-in USB controller in GPL162002A/162003A can be configured either as a USB device controller or USB mini-host controller.

AS a USB device, there are 4 endpoints: control pipe, bulk in, bulk out and interrupt in; furthermore, as a USB mini host, basic transaction and function are supported. There is only a FIFO implemented by 128X8 bits single port SRAM. To enhance the speed of bulk transfers, DMA function is supported and then FIFO seems to become dual FIFOs (each one is 64 bytes).

In addition, it is permissible that bulk in happens simultaneously with bulk out when DMA is disabled; this situation, however, is not allowed when DMA is enabled.

- Conforms to USB Version 1.1 specification
- USB device controller is supported
- USB mini-host controller is supported
- Built-in USB transceiver
- There are 4 endpoints when USB device is enabled
 - Control pipe for standard commands
 - Bulk IN for a large number of data transfers
 - Bulk OUT for a large number of data transfers
 - Interrupt in for data transfer seldom happens
- A 8-byte DFF FIFO for a control pipe in USB device only
- For USB device, a 128x8bits single port SRAM is used only in Bulk IN and Bulk OUT and for USB host, all types of transmission use this single port SRAM
- A 2-byte DFF FIFO for Interrupt IN in USB device only
- Functions are supported when USB mini-host is enabled
 - Setup command or data transaction
 - IN transaction / OUT transaction
 - Programmable packet delay time or timeout latency
 - SOF timer / frame number generator
 - Reset signal
- Interrupt mode or polling mode for driver

15.2 USB Device

When GPL162002A/162003A is used as a USB device, it supports 4 endpoints, control pipe, bulk in, bulk out, and interrupt in. For control pipe, when GPL162002A/162003A receives the standard command, it will automatically reply it except Get / Set Descriptor. That is, when it receives GET_STATUS, CLEAR_FEATURE, SET_FEATURE, SET_ADDRESS, GET_CONFIGURATION, SET_CONFIGURATION, GET_INTERFACE, and SET_INTERFACE, GPL162002A/162003A will auto reply these standard commands. For bulk in and bulk out, the maximum packet size is 64 bytes. GPL162002A/162003A supports non-DMA or DMA transfers. For non-DMA mode, it is 8 bits for MCU access and it is 16 bits access for DMA mode.

15.3 USB Mini-Host

GPL162002A/162003A can be used as a USB mini host. It supports commands, IN and OUT transfer. For command transfer, it is 8 bits for each data. For IN and OUT transfer, the maximum packet size is 64 bytes, and it is 8 bits for MCU access or 16 bits for DMA access. When it uses DMA mode, the data to be transferred must be multiple of 64 bytes.

15.4 Serial Interface Control Pin Configuration

Name	I/O	Description
DP	I/O	USB D+ pin
DN	I/O	USB D- pin

15.5 Control Registers

USB Device Register Summary Table

Name	Address	Description
P_USBD_Config	0x7B30	USB Configuration Register
P_USBD_Device	0x7B57	USB Device Register
P_USBD_Function	0x7B31	USB Function Register
P_USBD_DMAINT	0x7B59	USB DMA Interrupt Register
P_USBD_PMR	0x7B32	USB Power Management Register
P_USBD_EP0Data	0x7B33	USB Endpoint0 Data Register
P_USBD_BIData	0x7B34	USB Bulk In Data Register
P_USBD_BOData	0x7B35	USB Bulk Out Data Register
P_USBD_INTINData	0x7B36	USB Interrupt In Data Register
P_USBD_NullPkt	0x7B58	USB Null Packet Register
P_USBD_EPEvent	0x7B37	USB Endpoint Event Register

Name	Address	Description
P_USBD_GLOINT	0x7B38	USB Global Interrupt Register
P_USBD_INTEN	0x7B39	USB Interrupt Enable Register
P_USBD_INTF	0x7B3A	USB Interrupt Flag Register
P_USBD_SCI NTEN	0x7B3B	USB Standard Command Interrupt Enable Register
P_USBD_SCINTF	0x7B3C	USB Standard Command Interrupt Flag Register
P_USBD_EPAutoSet	0x7B3D	USB Endpoint Auto Set Register
P_USBD_EPSetStall	0x7B3E	USB Endpoint Set Stall Register
P_USBD_EPBufClear	0x7B3F	USB Endpoint Buffer Clear Register
P_USBD_EPEvntClear	0x7B40	USB Endpoint Event Clear Register
P_USBD_EP0WrtCount	0x7B41	USB Endpoint0 Write Count Register
P_USBD_BOWrtCount	0x7B42	USB Bulk Out Write Count Register
P_USBD_EP0BufPointer	0x7B43	USB Endpoint0 Buffer Pointer Register
P_USBD_BIBufPointer	0x7B44	USB Bulk In Buffer Pointer Register
P_USBD_BOBufPointer	0x7B45	USB Bulk Out Buffer Pointer Register
P_USBD_EP0RTR	0x7B46	USB Endpoint0 bmRequestType Register
P_USBD_EP0RR	0x7B47	USB Endpoint0 bRequest Register
P_USBD_EP0VR	0x7B48	USB Endpoint0 wValue Register
P_USBD_EP0IR	0x7B49	USB Endpoint0 wIndex Register
P_USBD_EP0LR	0x7B4A	USB Endpoint0 wLength Register
P_USBD_DMAWrtCountL	0x7B50	USB DMA Byte Count Low Register
P_USBD_DMAWrtCountH	0x7B51	USB DMA Byte Count High Register
P_USBD_DMAAckL	0x7B52	USB DMA ACK Count Low Register
P_USBD_DMAAckH	0x7B53	USB DMA ACK Count High Register
P_USBD_EPStall	0x7B54	USB Endpoint Stall Bit Register

USB Host Register Summary Table

Name	Address	Description
P_USBH_Config	0x7B00	USB Host Configuration Register
P_USBH_TimeConfig	0x7B01	USB Host Timing Configuration Register
P_USBH_Data	0x7B02	USB Host Data Register
P_USBH_Transfer	0x7B03	USB Host Transfer Register
P_USBH_DveAddr	0x7B04	USB Device Address Register
P_USBH_DveEP	0x7B05	USB Device Endpoint Register
P_USBH_TXCount	0x7B06	USB Host Transmit Count Register
P_USBH_RXCount	0x7B07	USB Receive Count Register
P_USBH_FIFOInPointer	0x7B08	USB Host FIFO Input Pointer Register
P_USBH_FIFOOOutPointer	0x7B09	USB Host FIFO Output Pointer Register

Name	Address	Description
P_USBH_AutoInByteCount	0x7B0A	USB Host Automatic In Transaction Byte Count Register
P_USBH_AutoOutByteCount	0x7B0B	USB Host Automatic Out Transaction Byte Count Register
P_USBH_AutoTrans	0x7B0C	USB Host Auto Transfer Register
P_USBH_Status	0x7B0D	USB Host Status Register
P_USBH_INTF	0x7B0E	USB Host Interrupt Flag Register
P_USBH_INTEN	0x7B0F	USB Host Interrupt Enable Register
P_USBH_StorageRST	0x7B10	USB Storage Reset Register
P_USBH_SoftRST	0x7B11	USB Software Reset Register / Device Plug Out Register
P_USBH_SOFTimer	0x7B12	USB SOF Timer Register
P_USBH_FrameNum	0x7B13	USB Frame Number Register
P_USBH_OTGConfig	0x7B14	USB OTG Configuration Register
P_USBH_VBusSet	0x7B15	USB VBUS Set Register
P_USBH_VbusStatus	0x7B16	USB VBUS Status Register
P_USBH_INAckCount	0x7B17	USB IN ACK Count Register
P_USBH_OutAckCount	0x7B18	USB OUT ACK Count Register
P_USBH_RSTAckCount	0x7B19	USB Reset ACK Count Register
P_USBH_Storage1/2	0x7B1A	For Debugging
P_USBH_DReadback	0x7B1B	USB D+ / D- Readback Register

15.6 USB Device Register Definition

P_USBD Config		0x7B30 USB Configuration Register															
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	RWUPEN	SPWR	USBEN	TNSPL	TNSPH	BYPASS
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	RWIPEN	R/W	Remote Wakeup Enable If this bit is set to "1", the remote wakeup is supported. Or, it is not supported.	0= Remote Wakeup is not supported 1= Remote Wakeup is supported
4	SPWR	R/W	Self Power of Device This bit is used to indicate if USB device is self-powered or not.	0= USB device is bus-powered 1= USB device is self-powered
3	USBEN	R/W	USB Transceiver Enable Write "1" to this bit to enable the USB transceiver.	

Bit	Function	Type	Description	Condition
2	TNSPL	R/W	USB Transceiver Pull Low Write "1" to this bit, D+ and D- are pulled down with 15K Ohm.	
1	TNSPH	R/W	USB Transceiver Pull High Write "1" to this bit, D+ is pulled high with 1.5K ohm.	
0	BYPASS	R/W	USB Bypass Mode	0= Bypass disable 1= Bypass enable, the inner USB transceiver is disabled.

P_USBD Device
0x7B57
USB Device Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP4_Type	EP3_Type	EP2_Type	EP1_Type	EP4_IO	EP3_IO	EP2_IO	EP1_IO	-	-	-	MOD				
Default	0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]	EP4_Type	R/W	Endpoint4 Type These two bits are used to indicate the type of endpoint4.	00 = Reserved 01 = Reserved 10 = Bulk 11 = Interrupt
[13:12]	EP3_Type	R/W	Endpoint3 Type These two bits are used to indicate the type of endpoint3.	00 = Reserved 01 = Reserved 10 = Bulk 11 = Interrupt
[11:10]	EP2_Type	R/W	Endpoint2 Type These two bits are used to indicate the type of endpoint2.	00 = Reserved 01 = Reserved 10 = Bulk 11 = Interrupt
[9:8]	EP1_Type	R/W	Endpoint1 Type These two bits are used to indicate the type of endpoint1.	00 = Reserved 01 = Reserved 10 = Bulk 11 = Interrupt
7	EP4_IO	R/W	Endpoint4 IN/OUT This bit is used to indicate the EP4 is in or out.	0 = OUT 1 = IN
6	EP3_IO	R/W	Endpoint3 IN/OUT This bit is used to indicate the EP3 is in or out.	0 = OUT 1 = IN
5	EP2_IO	R/W	Endpoint2 IN/OUT This bit is used to indicate the EP2 is in or out.	0 = OUT 1 = IN
4	EP1_IO	R/W	Endpoint1 IN/OUT This bit is used to indicate the EP1 is in or out.	0 = OUT 1 = IN
[3:0]			Reserved	

Bit	Function	Type	Description	Condition
1	MODE	R/W	Mode selection	0 = Normal mode 1 = Debug mode

P_USBD_Function
0x7B31
USB Function Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	SRST	DMA_BOEN	DMA_BIEN	Config_Value								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	SRST	W	Software Reset Write 1 to start USB software reset and programmer must write 0 to stop USB software reset.	1 : start software reset 0 : stop software reset
10	DMA_BOEN	R/W	DMA Bulk OUT Enable Write "1" to this bit to enable the DMA function with bulk out. When this bit is set to 1, the DMA_BIEN must be 0. It indicates Bulk IN must be disabled when Bulk out is enabled in DMA mode. When DMA is finished, this bit will be automatically cleared to "0".	Write 0= Diable DMA function with bulk out Write 1= Enable DMA function with bulk out
9	DMA_BIEN	R/W	DMA Bulk IN Enable Write "1" to this bit to enable the DMA function with bulk in. When this bit is set to 1, the DMA_BOEN must be 0. It indicates Bulk OUT must be disabled when Bulk in is enabled in DMA mode. When DMA is finished, this bit will be automatically cleared.	Write 0= Diable DMA function with bulk in Write 1= Enable DMA function with bulk in
[8:7]	Config_Value	R	Configure Value The USB configuration value of the device can be read from these two bits when receiving a configuration command.	
[6:0]	FNC_Addr	R	Function Address When the device gets "Set Address" command from the host, the address is stored in these bits.	

P_USBD_DMAINT 0x7B59
USB DMA Interrupt Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	DMAINTEN_CLR	DMAINTEN	DMAINTF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
2	DMAINTEN_CLR	R/W	DMA Interrupt Disable If this bit is set to "1", DMA Interrupt will be disabled.	Write 1 = DMA Interrupt Disable
1	DMAINTEN	R/W	DMA Interrupt Enable If this bit is set to "1", and a DAM interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. To select between IRQ3 and FIQ, please refer to Chapter Interrupt .	Write 1 = DMA Interrupt Enable
0	DMAINTF	R/W	DMA Interrupt Flag This bit is set if one of DMA interrupts happens. The interrupt indicates Bulk Out or Bulk In transaction is finished in DMA mode. Writing 1 to clear the specific interrupt would clear this bit.	Read 1 = Occurred Read 0 = Not occurred

P_USBD_PMR 0x7B32
USB Power Management Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	RESWKE	RE_WA	RE_WAFEA	RST	SUS_Mod
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
4	RESWKE	R/W	Resume Wakeup Enable Write "1" to enable "Resume Wakeup". (Returning from suspend mode indicates the system or CPU that USB is wakeup now). Write "0" to disable this function.	0: Disable 1: Enable

Bit	Function	Type	Description	Condition
3	RE_WA	R/W	Remote_Wakeup The USB will generate a resume signal of a duration of 0.24 micro-second when this bit is set in suspend mode and the USB device has already received the SET_FEATURE command of REMOTE_WAKEUP. This bit must be cleared manually after SUSPEND_MODE is cleared.	
2	RE_WAFEA	R	Remote Wakeup Feature When the device gets SET_FEATURE command of REMOTE_WAKEUP, this bit will set to 1 by hardware. When the device gets CLEAR_FEATURE command of REMOTE_WAKEUP, this bit will clear to 0 by hardware.	
1	RST	R	Reset This bit is assigned to reset USB-BUS. Logic 0 of this bit means there is a USB reset; on the contrary, logic 1 means no reset.	0: USB reset 1: No reset
0	SUS_Mod	R	Suspend Mode This bit is set by hardware when it enters suspend mode. This bit is cleared by hardware when it returns from suspend mode or the USB reset signal is generated.	

P_USBD_EP0Data
0x7B33
USB Endpoint0 Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	EP0DATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	EP0DATA	R/W	Endpoint0 Data	

P_USBD_BIData
0x7B34
USB Bulk IN Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	BIDATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	BIDATA	R/W	Bulk In Data	

P_USBD_BOData		0x7B35								USB Bulk OUT Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	BODATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	BODATA	R/W	Bulk Out Data	

P_USBD_INTINData		0x7B36								USB Interrupt IN Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	INTINDATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	INTINDATA	R/W	Interrupt IN Data	

P_USBD_NullPkt		0x7B58												USB Null Packet Register			
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	IIN_NULLPKT	BI_NULLPKT	EP0_NULLPKT
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
2	IIN_NULLPKT	R/W	Write "1" to enable USB device to send a null packet to the Interrupt IN endpoint. Write "0" to disable that function.	
1	BI_NULLPKT	R/W	Write "1" to enable USB device to send a null packet to the Bulk IN endpoint. Write "0" to disable that function.	
0	EP0_NULLPKT	R/W	Write "1" to enable USB device to send a null packet to the Endpoint0. Write "0" to disable that function.	

P_USBD_EPEvent		0x7B37								USB Endpoint Event Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IINNA	IINPR	BONA	BOPR	BOPE	BINA	BIPC	BIPR	E0SNA	E0SEN	E0INNA	E0INPR	E0ONA	E0OPR	E0OPE	E0SPR
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	IINNA	R/W	Interrupt IN NACK This bit is set to "1" if an IN request happens and in the meantime, the device sends a NAK. Write 1 to clear the bit.	Read 1= Occurred Read 0= Not occurred Write 0= No effect Write 1= Clear the flag
14	IINPR	R/W	Interrupt IN Packet Ready Set this bit to "1" to indicate an IN packet is ready in the INTERRUPT_IN FIFO. It's set automatically if there are 8 bytes IN data and P_USBD_EPAutoSet [4] is 1. After this bit is set to "1", the hardware will send the data to the Host. This bit is cleared by hardware when the data transaction is finished.	Write 1= Packet is ready
13	BONA	R/W	Bulk Out NACK This bit is set if an OUT packet happens and in the meantime, the device sends a NAK. Write 1 to clear this bit.	Read 1= Occurred Read 0= Not occurred Write 0= No effect Write 1= Clear the flag
12	BOPR	R/W	Bulk Out Packet Ready This bit is set if an OUT packet is loaded into the bulk out FIFO. Write 1 to clear this bit or it's cleared automatically if the OUT packet is read from MCU and P_USBD_EPAutoSet [3] is 1.	Read 1= Ready Read 0= Not ready Write 0= No effect Write 1= Clear the flag
11	BOPE	R/W	Bulk Out Packet Enable Write "1" to this bit to enable receiving incoming packets for BULK OUT data. This bit is automatically cleared after the packet is loaded to bulk out FIFO in DMA mode or BOPR is set to "1". Writing "1" to P_USBD_EPEvntClear [4] will clear this bit.	Write 1= Enable
10	BINA	R/W	Bulk IN NACK This bit is set if an IN request happens but the device sends a NAK. Write 1 to clear this bit.	Read 1= Occurred Read 0= Not occurred Write 0= No effect Write 1= Clear the flag
9	BIPC	R/W	Bulk IN Packet Clear This bit is set if an IN packet is read from the host. Write 1 to clear this bit.	Read 1= Occurred Read 0= Not occurred Write 0= No effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
8	BIPR	R/W	<p>Bulk IN Packet Ready</p> <p>Set this bit to indicate BULK IN packet is ready in the BULK_IN FIFO. It's set automatically if MCU writes 64 bytes data and P_USBD_EPAutoSet [2] is 1. After this bit is set to "1", the hardware will send the data to the Host.</p> <p>This bit is cleared by hardware when the data transaction is finished.</p>	Write 1= Packet is ready
7	E0SNA	R/W	<p>EP0 Status NACK</p> <p>This bit is set if the request of status transaction happens but the device sends a NAK. Wirte 1 to clear the bit.</p>	<p>Read 1= Occurred</p> <p>Read 0= Not occurred</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>
6	E0SEN	R/W	<p>EP0 Status Enable</p> <p>This bit is set to enable the transaction in status stage. It's automatically cleared if the status stage is finished. Besides, it's set automatically if the status stage is finished and P_USBD_EPAutoSet [5] is 1.</p>	Write 1= Enable
5	E0INNA	R/W	<p>EP0 IN NACK</p> <p>This bit is set if an IN request happens but the device sends a NAK. Wirte 1 to clear the bit.</p>	<p>Read 1= Occurred</p> <p>Read 0= Not occurred</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>
4	E0INPR	R/W	<p>EP0 IN Packet Ready</p> <p>Set this bit to indicate IN packet is ready in the Endpoint0 FIFO. It's set automatically if there are 8 bytes IN data and P_USBD_EPAutoSet [1] is 1. After this bit is set to "1", the hardware will send the data to the Host.</p> <p>This bit is cleared by hardware when the data transaction is finished.</p>	Write 1 to indicate IN packet is ready in Endpoint0 FIFO.
3	E0ONA	R/W	<p>EP0 Out NACK</p> <p>This bit is set if an OUT packet happens but the device sends a NAK. Wirte 1 to clear the bit.</p>	<p>Read 1= Occurred</p> <p>Read 0= Not occurred</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>
2	E0OPR	R/W	<p>EP0 Out Packet Ready</p> <p>This bit is set if an OUT packet is loaded into the endpoint0 FIFO. Write 1 to clear the bit or it's cleared automatically if the OUT packet is read from MCU and P_USBD_EPAutoSet [0] is 1.</p>	<p>Read 1= Ready</p> <p>Read 0= Not ready</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>

Bit	Function	Type	Description	Condition
1	E0OPE	R/W	EP0 Out Packet Enable Writing "1" to this bit will enable the incoming packet for OUT data. This bit is automatically cleared after the packet is loaded to endpoint0 FIFO (E0OPR is set to "1"). Write "1" to P_USBD_EPEvtClear [0] will clear this bit.	Write 1= Enable
0	E0SPR	R/W	EP0 Setup Packet Ready This bit is set if a non-standard setup command or a get/set descriptor command is loaded into the endpoint0 FIFO. Write 1 to clear the bit.	Read 1= Occurred Read 0= Not occurred Write 1= Clear Write 0= No effect

P_USBD_GLOINT
0x7B38
USB Global Interrupt Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	DMA	STANDARD	POWER	INT	BO	BI	EP0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	DMA	R	DMA Interrupt This bit is set if one of DMA interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred
5	STANDARD	R	Standard Command Interrupt This bit is set if one of standard command interrupts happens, except GET/SET Descriptor. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred
4	POWER	R	Power Management Interrupt This bit is set if one of power management interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred
3	INT	R	Interrupt In interrupt This bit is set if one of INTERRUPT_IN interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred
2	BO	R	Bulk Out Interrupt This bit is set if one of BULK_OUT interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred

Bit	Function	Type	Description	Condition
1	BI	R	Bulk In Interrupt This bit is set if one of BULK_IN interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred
0	EP0	R	Endpoint0 Interrupt This bit is set if one of endpoint0 interrupts happens. Writing 1 to clear the specific interrupt would also clear this bit.	Read 1= Occurred Read 0= Not occurred

P_USBD_INTEN
0x7B39
USB Interrupt Enable Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RST	R/W	Rest Interrupt Enable If this bit is set to "1" and an interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
14	RME	R/W	Resume Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
13	SUS	R/W	Suspend Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
12	IINNA	R/W	Interrupt In NACK Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
11	IINPC	R/W	Interrupt In Packet Clear Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
10	BONA	R/W	Bulk Out NACK Interrupt Enable. If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable

Bit	Function	Type	Description	Condition
9	BOPS	R/W	Bulk Out Packet Set Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
8	BINA	R/W	Bulk In NACK Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
7	BIPC	R/W	Bulk In Packet Clear Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
6	E0SNA	R/W	EP0 Status NACK Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
5	E0SC	R/W	EP0 Status Clear Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
4	E0INNA	R/W	EP0 In NACK Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
3	E0INPC	R/W	EP0 In Packet Clear Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
2	E0ONA	R/W	EP0 Out NACK Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
1	E0OPS	R/W	EP0 Out Packet Set Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
0	E0SPS	R/W	EP0 Setup Packet Set Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable

P_USBD_INTF			0x7B3A				USB Interrupt Flag Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	RST	R/W	Reset Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and the USB device is reset, the interrupt flag will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	RME	R/W	Resume Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and the USB device resumes in SUSPEND state, the interrupt flag will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
13	SUS	R/W	Suspend Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and the USB device is suspended, the interrupt flag will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
12	IINNA	R/W	Interrupt In NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN request happens with rplying a NAK to the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
11	IINPC	R/W	Interrupt In Packet Clear Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN packet is read from the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
10	BONA	R/W	Bulk Out NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an OUT packet happens but the device sends a NAK, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
9	BOPS	R/W	Bulk Out Packet Set Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an OUT packet is loaded into the endpoint0 FIFO, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
8	BINA	R/W	Bulk In NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN request happens with rplying a NAK to the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
7	BIPC	R/W	Bulk In Packet Clear Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN packet is read from the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
6	E0SNA	R/W	EP0 Status NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and the request of status transaction happens with rplying a NAK to the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
5	E0SC	R/W	EP0 Status Clear Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and the status stage is finished, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
4	E0INNA	R/W	EP0 In NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN request happens with rplying a NAK to the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
3	E0INPC	R/W	EP0 In Packet Clear Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an IN packet is read from the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
2	E0ONA	R/W	EP0 Out NACK Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an OUT request happens with rplying a NAK to the host, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
1	E0OPS	R/W	EP0 Out Packet Set Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and an OUT packet is loaded into the endpoint0 FIFO, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
0	E0SPS	R/W	EP0 Setup Packet Set Interrupt Flag If the corresponding enable bit of P_USBD_INTEN is 1 and a non-standard setup command or get/set descriptor command is loaded into the endpoint0 FIFO, this bit will be set.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

P_USBD_SCINTEN				0x7B3B				USB Standard Command Interrupt Enable Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
7	GSTS	R/W	Get Status Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
6	CFEA	R/W	Clear Feature Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
5	SFEA	R/W	Set Feature Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
4	SADD	R/W	Set Address Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
3	GCON	R/W	Get Configuration Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
2	SCON	R/W	Set Configuration Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
1	GINT	R/W	Get Interface Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable
0	SINT	R/W	Set Interface Interrupt Enable If this bit is set to "1" and interrupt occurs, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked.	0= Disable 1= Enable

P_USBD_SCINTF	0x7B3C								USB Standard Command Interrupt Flag Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
7	GSTS	R/W	Get Status Interrupt Flag The interrupt is set if the enable bit is 1 and GET_STATUS command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
6	CFEA	R/W	Clear Feature Interrupt Flag The interrupt is set if the enable bit is 1 and CLEAR_FEATURE command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
5	SFEA	R/W	Set Feature Interrupt Flag The interrupt is set if the enable bit is 1 and SET_FEATURE command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
4	SADD	R/W	Set Address Interrupt Flag The interrupt is set if the enable bit is 1 and SET_ADDRESS command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
3	GCON	R/W	Get Configuration Interrupt Flag The interrupt is set if the enable bit is 1 and GET_CONFIGURATION command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
2	SCON	R/W	Set Configuration Interrupt Flag The interrupt is set if the enable bit is 1 and SET_CONFIGURATION command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
1	GINT	R/W	Get Interface Interrupt Flag The interrupt is set if the enable bit is 1 and GET_INTERFACE command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
0	SINT	R/W	Set Interface Interrupt Flag The interrupt is set if the enable bit is 1 and SET_INTERFACE command happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

P_USBD_EPAutoSet 0x7B3D
USB Endpoint Auto Set Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	EP0ASE	IAINPR	BAOPE	BAIPR	E0AIPR	E0AOPE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	EP0ASE	R/W	EP0 Auto Status Enable Set Please refer to P_USBD_EPEvent[6]	
4	IAINPR	R/W	Interrupt Auto In Packet Ready Set Please refer to P_USBD_EPEvent[14]	
3	BAOPE	R/W	Bulk Auto Out Packet Enable Set Please refer to P_USBD_EPEvent[11] or [12]	
2	BAIPR	R/W	Bulk Auto In Packet Ready Set Please refer to P_USBD_EPEvent[8]	
1	E0AIPR	R/W	EP0 Auto In Packet Ready Set Please refer to P_USBD_EPEvent[4]	
0	E0AOPE	R/W	EP0 Auto Out Packet Enable Set Please refer to P_USBD_EPEvent[1] or [2]	

P_USBD_EPSetStall 0x7B3E
USB Endpoint Set Stall Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	IINS	BOS	BIS	EP0S
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	IINS	R/W	Interrupt In Set Stall	1 = enable STALL response 0 = disable STALL response
2	BOS	R/W	Bulk Out Set Stall	1 = enable STALL response 0 = disable STALL response
1	BIS	R/W	Bulk In Set Stall	1 = enable STALL response 0 = disable STALL response
0	EP0S	R/W	EP0 Set Stall Write 1 to this bit to generate a STALL signal if a host request happens. This bit is automatically cleared if one of SETUP commands is loaded into endpoint0 FIFO.	1 = enable STALL response 0 = disable STALL response

P_USBD_EPBufClear 0x7B3F			USB Endpoint Buffer Clear Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	IINBC	BOBC	BIBC	EP0BC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	IINBC	W	Interrupt In Buffer Clear Write "1" to clear Interrupt In buffer	
2	BOBC	W	Bulk Out Buffer Clear Write "1" to clear Bulk Out buffer	
1	BIBC	W	Bulk In Buffer Clear Write "1" to clear Bulk In buffer	
0	EP0BC	W	EP0 Buffer Clear Write "1" to clear EP0 buffer	

P_USBD_EPEvtClear 0x7B40			USB Endpoint Event Clear Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	IINPC	BOEC	BIPC	EP0SC	EP0IPC	EP0OEC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	IINPC	W	Interrupt In Packet Clear Write "1" to this bit to clear P_USBD_EPEvent[14]	
4	BOEC	W	Bulk Out Enable Clear Write "1" to this bit to clear P_USBD_EPEvent[11]	
3	BIPC	W	Bulk In Packet Clear Write "1" to this bit to clear P_USBD_EPEvent[9]	
2	EP0SC	W	EP0 Status Clear Write "1" to this bit to clear P_USBD_EPEvent[6]	
1	EP0IPC	W	EP0 In Packet Clear Write "1" to this bit to clear P_USBD_EPEvent[4]	
0	EP0OEC	W	EP0 Out Enable Clear Write "1" to this bit to clear P_USBD_EPEvent[1]	

P_USBD_EP0WrtCount 0x7B41			USB Endpoint0 Write Count Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	EP0WC		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	EP0WC	R	EP0 Write Count Read these bits to acquire the number of data in the EP0 FIFO.	

P_USBD_BOWrCount 0x7B42 USB Bulk OUT Write Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	BOWC						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
[6:0]	BOWC	R	Bulk Out Write Count Read these bits to acquire the number of data in the Bulk Out FIFO.	

P_USBD_EP0BufPointer 0x7B43 USB Endpoint0 Buffer Pointer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	EP0WBP			EP0RBP		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
[5:3]	EP0WBP	R	EP0 Write Buffer Pointer These bits are used as the write pointer of EP0 FIFO.	
[2:0]	EP0RBP	R	EP0 Read Buffer Pointer These bits are used as the read pointer of EP0 FIFO.	

P_USBD_BIBufPointer 0x7B44 USB Bulk IN Buffer Pointer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BIBWP								BIBRP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	BIBWP	R	Bulk IN Buffer Write Pointer These bits are used as the write pointer of Bulk IN FIFO.	
[7:0]	BIBRP	R	Bulk IN Buffer Read Pointer These bits are used as the read pointer of Bulk IN FIFO.	

P_USBD BOBufPointer 0x7B45
USB Bulk OUT Buffer Pointer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BOBWP								BOBRP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]	BOBWP	R	Bulk OUT Buffer Write Pointer These bits are used as the write pointer of Bulk OUT FIFO.	
[7:0]	BOBRP	R	Bulk OUT Buffer Read Pointer These bits are used as the read pointer of Bulk OUT FIFO.	

P_USBD EP0RTR 0x7B46
USB Endpoint0 bmRequestType Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	EP0RTR							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	EP0RTR	R	EP0 bmRequestType These bits are used as the "Request Type" of setup commands.	

P_USBD EP0RR 0x7B47
USB Endpoint0 bRequest Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	EP0RR							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	EP0RR	R	EP0 bRequest These bits are used as the Request of setup commands.	

P_USBD EP0VR 0x7B48
USB Endpoint0 wValue Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP0VR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	EP0VR	R	EP0 wValue These bits are used as the Value of setup commands.	

P_USBD_EP0IR 0x7B49 USB wIndex Register			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																		
Function	EP0IR																	
Default	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																	

Bit	Function	Type	Description	Condition
[15:0]	EP0IR	R	EP0 wIndex These bits are used as the wIndex of setup commands.	

P_USBD_EP0LR 0x7B4A USB Endpoint0 wLength Register			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																		
Function	EP0LR																	
Default	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																	

Bit	Function	Type	Description	Condition
[15:0]	EP0LR	R	EP0 wLength These bits are used as the wLength of setup commands.	

P_USBD_DMAWrtCountL 0x7B50 USB DMA Byte Count Low Register			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																		
Function	DMAWCL																	
Default	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																	

Bit	Function	Type	Description	Condition
[15:0]	DMAWCL	R/W	DMA Write Count Low The register is used in DMA mode. Set this register to indicate how many data is transferred. (It's must be done after setting DMAWCH)	

P_USBD_DMAWrtCountH 0x7B51 USB DMA Byte Count High Register			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																		
Function	-	-	-	-	-	-	-	-	-	-	DMAWCH							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	DMAWCH	R/W	DMA Write Count High The register is used in DMA mode. Set this register to indicate how many data is transferred.	

P_USBD_DMAACK				0x7B52				USB DMA ACK Count Low Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DMAACKL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DMAACKL	R/W	DMA ACK Count Low The register is used in DMA mode. It indicates how many transactions (each transaction is a 64-byte packet) are not finished yet.	Write this port to reset DMAWC and DMAACK

P_USBD_DMAACK				0x7B53				USB DMA ACK Count High Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	DMAACKH					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
[2:0]	DMAACKH	R	DMA ACK Count High The register is used in DMA mode. It indicates how many transactions (each transaction is a 64-byte packet) are not finished yet.	

P_USBD_EPStall		0x7B54		USB Endpoint Stall Bit Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	IISS	BOSS	BISS	EP0SS	-	-	-	-	IISB	BOSB	BISB	EP0SB	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	IISS	R/W	Interrupt IN It indicates that USB device has replied the request by sending STALL.	Write "1" to clear this bit
10	BOSS	R/W	Bulk OUT It indicates that USB device has replied the request by sending STALL.	Write "1" to clear this bit
9	BISS	R/W	Bulk IN It indicates that USB device has replied the request by sending STALL.	Write "1" to clear this bit

Bit	Function	Type	Description	Condition
8	EP0SS	R/W	Endpoint0 It indicates that USB device has replied the request by sending STALL.	Write "1" to clear this bit
[7:4]			Reserved	
3	IISB	R	Interrupt IN If STALL happens, the bit will be set. The bit is for debugging only.	
2	BOSB	R	Bulk OUT If STALL happens, the bit will be set. The bit is for debugging only.	
1	BISB	R	Bulk IN If STALL happens, the bit will be set. The bit is for debugging only.	
0	EP0SB	R	Endpoint0 If STALL happens, the bit will be set. The bit is for debugging only.	

PS: In CPU mode, BULK_IN and BULK_OUT can work simultaneously. BULK_IN will occupy the USB buffer, 0x00~0x3F of SRAM, and BULK_OUT will occupy the USB buffer, 0x40~0x7F of SRAM.

15.7 USB Host Register Definition

Bit 0 in [P_USBH_Config] must be set to 1 when accessing the USB Host functions.

P_USBH_Config	0x7B00 USB Host Configuration Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	SUS	-	ASOF	SOFTR	-	HOSTEN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	SUS	R/W	Host Suspend Write "1" to this bit will enable host suspend.	0 = HOST SUSPEND is disable. 1 = HOST SUSPEND is enable. USB Transceiver is in SUSPEND mode. (The SUSPEND mode is controlled by software when HOST is enable.)
4			Reserved	
3	ASOF	R/W	Auto Generate SOF Setting this bit to "1" will generate SOF timer by hardware. If this bit is 0, write 1 to P_USBH_Transfer [0] to generate	0= Generate SOF by software 1= Generate SOF by hardware

Bit	Function	Type	Description	Condition
			SOF timer by software.	
2	SOFTTR	R/W	SOF Timer Write "1" to enable SOF timer.	0= Disable SOF timer 1= Enable SOF timer
1			Reserved	
0	HOSTEN	R/W	Host Enable Write "1" to this bit to enable host.	0= Host is disabled 1= Host is enabled (Device is disabled)

P_USBH_TimeConfig 0x7B01 USB Host Timing Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	SAU	PAC	TC		IPD	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:6]			Reserved	
5	SAU	R/W	Storage 1/2 Auto Mode If this bit is set to 1, setting P_USBH_StorageRST after each transaction in non-DMA mode is not necessary.	
4	PAC	R/W	Pointer Auto Clear If the bit is set 1, READ/WRITE pointer is automatically reset to 0 after any transaction. However, this mode is automatically disabled in DMA mode. The DMA mode is activated by configuring P_USBH_AutoTrans [0] or P_USBH_AutoTrans [1].	Always set to 1
[3:2]	TC	R/W	TimeOut Criteria	00= 16T 01= 18T 10= 20T 11= 22T
[1:0]	IPD	R/W	Inter Packet Delay	00= 2T 01= 4T 10= 6T 11= 8T

P_USBH_Data 0x7B02 USB Host Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	HDATA	R/W	Host Data	

P_USBH_Transfer
0x7B03
USB Host Transfer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	RST	OD1	OD0	ID1	ID0	Setup	SOF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	RST	R/W	Reset Signal The RESET signaling is not terminated by hardware automatically; CPU must explicitly write 0 to this bit to stop signaling RESET instead.	1= Generate reset signal 0= Stop reset signal
5	OD1	R/W	Out Data1 Transfer Write 1 to initiate OUT DATA1 transfer. This bit is cleared automatically if the transfer is completed.	
4	OD0	R/W	Out Data0 Transfer Write 1 to initiate OUT DATA0 transfer. This bit is cleared automatically if the transfer is completed.	
3	ID1	R/W	In Data1 Transfer Write 1 to initiate IN DATA1 transfer. This bit is cleared automatically if the transfer is completed.	
2	ID0	R/W	In Data0 Transfer Write 1 to initiate IN DATA0 transfer. This bit is cleared automatically if the transfer is completed.	
1	SETUP	R/W	Setup Transfer Write 1 to initiate SETUP transfer. This bit is cleared automatically if the transfer is completed.	
0	SOF	R/W	SOF Transfer Write 1 to initiate SOF transfer. (SOF is generated by software) This bit is cleared automatically if the transfer is completed.	

P_USBH_DveAddr
0x7B04
USB Device Address Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	DAddr						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	

Bit	Function	Type	Description	Condition
[6:0]	Daddr	R/W	Device Address Write the device address to these bits that host wants to access.	

P_USBH_DveEP 0x7B05 USB Device Endpoint Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	DEP			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
[3:0]	DEP	R/W	Device Endpoint Write the device endpoint to these bits that host wants to access.	

P_USBH_TXCount 0x7B06 USB Host Transmit Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	TXC			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
[6:0]	TXC	R/W	Host Transmitting Count Write the number of data to these bits that host wants to transmit.	

P_USBH_RXCount 0x7B07 USB Host Receive Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	RXC			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
[6:0]	RXC	R/W	Host Receiving Count The number of data that host receives is stored in these bits.	

P_USBH_FIFOInPointer 0x7B08 USB Host FIFO Input Pointer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	HFIP			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	HFIP	R/W	Host FIFO Input Pointer	

P_USBH_FIFOPutPointer 0x7B09

USB Host FIFO Output Pointer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	HFOP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	HFOP	R/W	Host FIFO Output Pointer	

P_USBH_AutoInByteCount

0x7B0A

USB Host Automatic In Transaction Byte Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HAIBC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	HAIBC	R/W	Host Automatic IN Transaction Byte Count Write the number of IN transaction that is to be initiated in these bits. For example, if the host is going to receive 512 bytes from the device, it has to generate $512/64=8$ IN transactions, and write 8 to this register. This register is used only in DMA mode.	When using DMA in USB host, the transmitting/receiving data must be multiple of 64 bytes.

P_USBH_AutoOutByteCount

0x7B0B

USB Host Automatic Out Transaction Byte Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HAOBC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	HAOBC	R/W	Host Automatic OUT Transaction Byte Count Write the number of OUT transaction in these bits that is to be initiated. For example, if the host is going to transmit 512 bytes to the device, it has to generate $512/64=8$ OUT	When using DMA in USB host, the transmit / receive byte must be multiple of 64 bytes.

Bit	Function	Type	Description	Condition
			transactions, and write 8 to this register. This register is used only in DMA mode.	

P_USBH_AutoTrans
0x7B0C
USB Host Auto Transfer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	CAO	CAI	AOX	AIX
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]			Reserved	
3	CAO	W	Clear Auto Out Transfer Write "1" to this bit to clear P_USBH_AutoTrans[1]	Write 0 = Not active Write 1 = clear
2	CAI	W	Clear Auto IN Transfer Write "1" to this bit to clear P_USBH_AutoTrans[0]	Write 0 = Not active Write 1 = clear
1	AOX	R/W	Auto Out Transfer This bit is set 1 for entering DMA mode for Bulk out. When transaction is surely finished, this bit must be cleared. Note: After DMA is finished (DMA counter reaches 0), the transaction may not be finished yet. Enabling this bit, programming the UHAOBCR and then triggering the OUT transfer will start the OUT transaction. This bit is cleared by software.	0: Disable 1: Enable
0	AIX	R/W	Auto In Transfer This bit is set 1 for entering DMA mode for Bulk in. When transaction is surely finished, this bit must be cleared. Enabling this bit, programming the UHAIBCR and then triggering the IN transfer will start the IN transaction. This bit is cleared by software.	0: Disable 1: Enable

P_USBH_Status
0x7B0D
USB Host Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	TO	CRC	DE	BS	UP	SH	NH	AH
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	

Bit	Function	Type	Description	Condition
7	TO	R/W	TimeOut, No Response This flag is set when timeout or no response occurs from the device.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
6	CRC	R/W	CRC16 Error Packet Received This flag is set when the CRC16 error occurs in received data packet.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
5	DE	R/W	Data Sequence Error Packet Received This flag is set when the data sequence error occurs in received data packet.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
4	BS	R/W	Bit Stuffing Error Packet Received This flag is set when the bit stuffing error occurs in received data packet.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
3	UP	R/W	Unkonwn PID Packet Received This flag is set when receiving a packet with unknown PID.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
2	SH	R/W	Stall Handshake Received This flag is set when receiving a stall handshake.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
1	NH	R/W	NACK Handshake Received This flag is set when receiving a NACK handshake.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
0	AH	R/W	ACK Handshake Received This flag is set when receiving an ACK handshake.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

P_USBH_INTF
0x7B0E
USB Host Interrupt Flag Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	

Bit	Function	Type	Description	Condition
11	DPO	R/W	Device Plug Out Interrupt This interrupt is asserted whenever host has detected device is plug-out. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
10	TRST	R/W	Transmit USB Reset Interrupt This interrupt is asserted whenever host has sent USB RESET signal. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
9	TSOFI	R/W	Transmit SOF Interrupt This interrupt is asserted whenever host has sent a SOF. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
8	ITOK	R/W	IN Token Transmit Interrupt This interrupt is asserted whenever host has sent an IN Token. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
7	TXO	R/W	Transmit Data Interrupt This interrupt is asserted whenever host has sent a DATA packet. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
6	VSC	R/W	VBUS Status Change Interrupt This interrupt is asserted whenever VBUS status has been changed.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
5	AOX	R/W	Automatic Out Transfer Interrupt This interrupt is asserted when the host has transmitted the data out to the device or host has received an ACK from the device.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
4	AIX	R/W	Automatic In Transfer Interrupt	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
3	RX	R/W	Receive Interrupt This interrupt is asserted whenever the host receives a packet form the device. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
2	TX	R/W	Transmit Interrupt This interrupt is asserted whenever the transmitting task is completed. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
1	SOF	R/W	SOF Interrupt This interrupt is periodically generated for every 1ms frame time. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
0	DSC	R/W	DP DM Status Change Interrupt This interrupt is used to detect the device connection when the host controller is in the idle state. Once the device is plug-in, this interrupt must be disabled. Write 1 to clear the interrupt.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag

P_USBH_INTEN
0x7B0F
USB Host Interrupt Enable Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	DPO	R/W	Device Plug Out Interrupt Enable	0= Disable 1= Enable
10	TRST	R/W	Transmit USB Reset Interrupt Enable	0= Disable 1= Enable
9	TSOFI	R/W	Transmit SOF Interrupt Enable	0= Disable 1= Enable
8	ITOK	R/W	IN Token Transmit Interrupt Enable	0= Disable 1= Enable
7	TXO	R/W	Transmit Data Interrupt Enable	0= Disable 1= Enable
6	VSC	R/W	VBUS Status Change Interrupt Enable	0= Disable 1= Enable
5	AOX	R/W	Automatic Out Transfer Interrupt Enable	0= Disable 1= Enable
4	AIX	R/W	Automatic In Transfer Interrupt Enable	0= Disable 1= Enable
3	RX	R/W	Receive Interrupt Enable	0= Disable 1= Enable
2	TX	R/W	Transmit Interrupt Enable	0= Disable 1= Enable
1	SOF	R/W	SOF Interrupt Enable	0= Disable 1= Enable

Bit	Function	Type	Description	Condition
0	DSC	R/W	DP DM Status Change Interrupt Enable	0= Disable 1= Enable

P_USBH_StorageRST
0x7B10
USB Storage Reset Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	StorageRST															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	StorageRST	W	Reset Bulk In/Out Buffer Write any value to this register will clear bulk in/out buffer. We suggest programmer write this register at starting data transmission with USB device.	Write any value to reset bulk in/out buffer

P_USBH_SoftRST
0x7B11
USB Software Reset Register / Device Plug Out Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	DPOE	DPOTV						SRST	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:9]			Reserved	
8	DPOE	R/W	Device Plug Out Timer Enable Write "1" to this bit to enable the timer.	
[7:1]	DPOTV	R/W	Device Plug Out Timer Value If DEVICE_PLUG_OUT_TIMER_ENABLE is 1, the inside timer is enabled. For each clock cycle, if D+ and D- are all 0, the timer is added by 1. Besides, if one of D+/D- is not 0, the timer is reset to 0. When it counts to DEVICE_PLUG_OUT_TIMER_VALUE, an interrupt is happened and DEVICE_PLUG_TIMER_OUT_ENABLE is reset.	
0	SRST	R/w	Software Reset	

P_USBH_INAckCount
0x7B17
USB IN ACK Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	INACK															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	INACK	R/W	IN ACK Count	

P_USBH_OutAckCount 0x7B18
USB OUT ACK Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	OUTACK															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	OUTACK	R/W	OUT ACK Count	

P_USBH_RSTAckCount 0x7B19
USB Reset ACK Count Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IARST	OARST
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:2]			Reserved	
1	IARST	W	IN ACK Reset Write "1" to reset P_USBH_INAckCount	
0	OARST	W	OUT ACK Reset Write "1" to reset P_USBH_OUTAckCount	

P_USBH_Dreadback 0x7B1B
USB D+ / D- Readback Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DM	DP
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:2]			Reserved	
1	DM	R	VPIN input signal	
0	DP	R	VMIN input signal	

16 Analog Input (Touch Panel / Voice Recorder)

16.1 Introduction

There are two ADC modules on GPL162002A/162003A. One is a 12-bit SAR ADC for touch panel and voltage detection, and the other one is a 16-bit voice ADC (so called HQADC) for voice recording. The SAR ADC only supports manual sampling mode whereas the HQADC only supports auto sampling mode.

- 6 Channels, 12-bit resolution (11-bit no-missing code) ADC for touch panel and voltage detection.
- 80dB delta sigma ADC for voice recording and microphone input.
 - 16-bit stereo ADC of 48 KHz sampling rate for audio-band signal-processing applications.
 - On-chip microphone bias-voltage output.
 - On-chip microphone boost amplifier (gain=0 or 20dB).
 - On-chip programmable gain amplifier for MIC input. (gain=33, 31.5, 30, ~ -12, -∞ dB)
 - On-chip volume control for line-in & FM-in. (gain= 12, 10.5, 9, ~ -33, -8 dB)
 - On-chip anti-aliasing filter.
 - Analog oversampling third-order sigma-delta modulator for ADC.
 - On-chip digital decimation comb filter & decimation FIR filters.
- 12.288MHz master clock frequency for ADC.

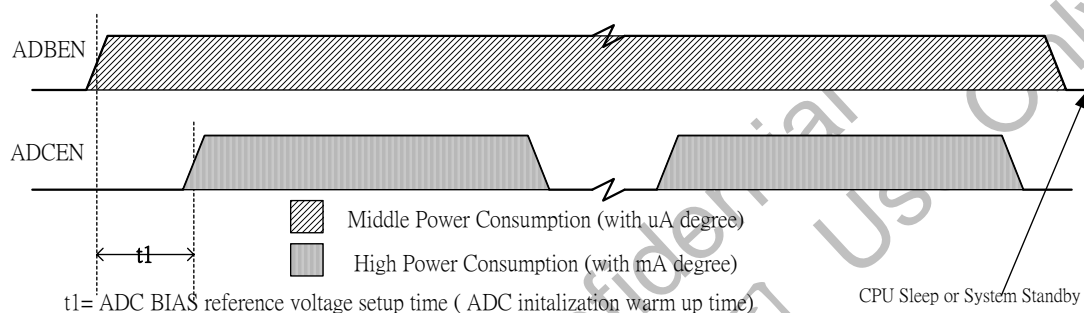
16.2 SAR ADC Control

Six channels of the 12-bit SAR ADC are built in GPL162002A/162003A. Two channels are dedicated for touch panel, named TP channels; other four channels are defined as general-purpose line input LINEIN1, LINEIN2, LINEIN3 and LINEIN4. These four channels are very suitable for system voltage detection and other general-purpose usage.

GPL162002A/162003A SAR AD conversion process is done automatically by hardware. Initializing and enabling ADC is two necessary steps to make GPL162002A/162003A ADC work properly. SAR ADC is unable to work without being initialized even when ADC is enabled and vice versa. Initialing SAR ADC is to build up internal reference voltage for itself. It takes approximate 100 milliseconds (ms) to complete the initialization. After initialization, GPL162002A/162003A power consumption will increase several hundreds of micro-amp (uA). On the other hand, enabling ADC is to turn the SAR ADC module on. It takes only several microseconds (us), but it will consume more than 2mA. There are two control bits to Initialize and enable SAR ADC correspondingly.

To increase SAR AD conversion efficiency, SAR ADC must not be initialized (to build up internal reference voltage) right before AD conversion operation. Instead, SAR ADC should be initialized for a

while before starting ADC data acquirement because of its long setup time. In touch panel applications, Generalplus recommends initializing SAR ADC right after power on or after sleep / halt mode. After all, system requires SAR ADC to sample the x, y coordinate value continuously on this case. There is one control bit to turn off SAR ADC internal reference voltage by software (set up by initialization operation). Programmers should shut down SAR ADC internal reference voltage before GPL162002A/162003A enters sleep mode or halt mode. Moreover, programmers should re-initialize SAR ADC again after power on and after wakeup from sleep / halt mode if necessary.



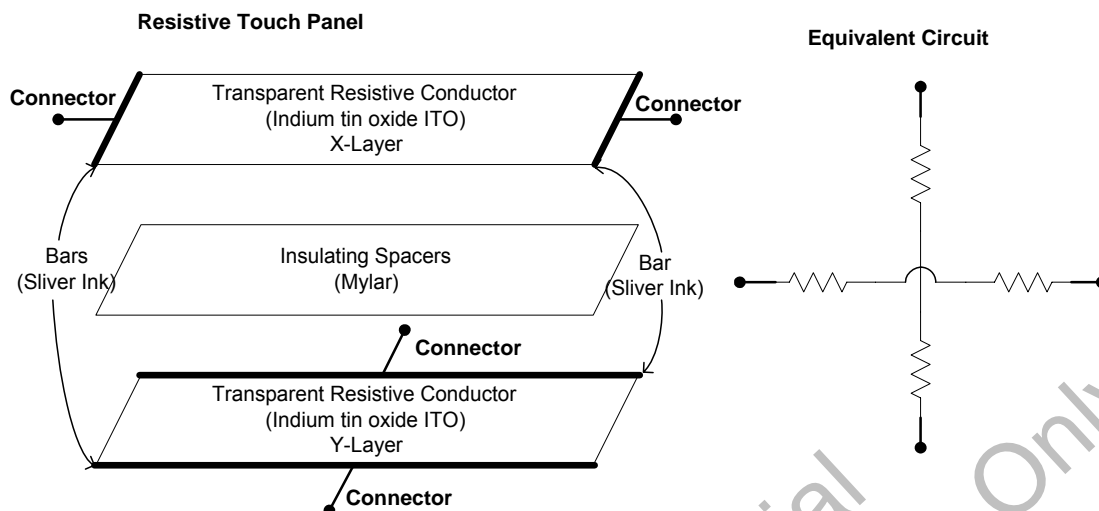
Note that while SAR ADC is turned on, there will be larger power consumption. Therefore, the time interval when SAR ADC is enabled should be as short as possible. Generalplus recommends enabling SAR ADC right before SAR ADC channel sample operation and turn off SAR ADC immediately after ADC data acquirement. However, in order to ease the software effect and decrease the CPU bandwidth, programmers might always turn on ADC during voice recording.

Note that AD Conversion time is programmable and can be set as SYSCLK/2048, SYSCLK/1024, SYSCLK/512, SYSCLK/256, SYSCLK/128, and SYSCLK/64 for using ADC under different kind of SYSCLK frequency.

16.3 Touch Panel Interface

GPL162002A/162003A supports analog touch panel interface. It includes two dedicated SAR ADC channels, some built-in switches and de-bounces circuit. In addition, GPL162002A/162003A has interrupt and wakeup mechanisms while a stylus taps on the touch panel. It also allows programmers to poll the touch panel status for checking the stylus tapping. There are only four pins for the touch panel interface: TSPX, TSMX, TSPY, and TSMY.

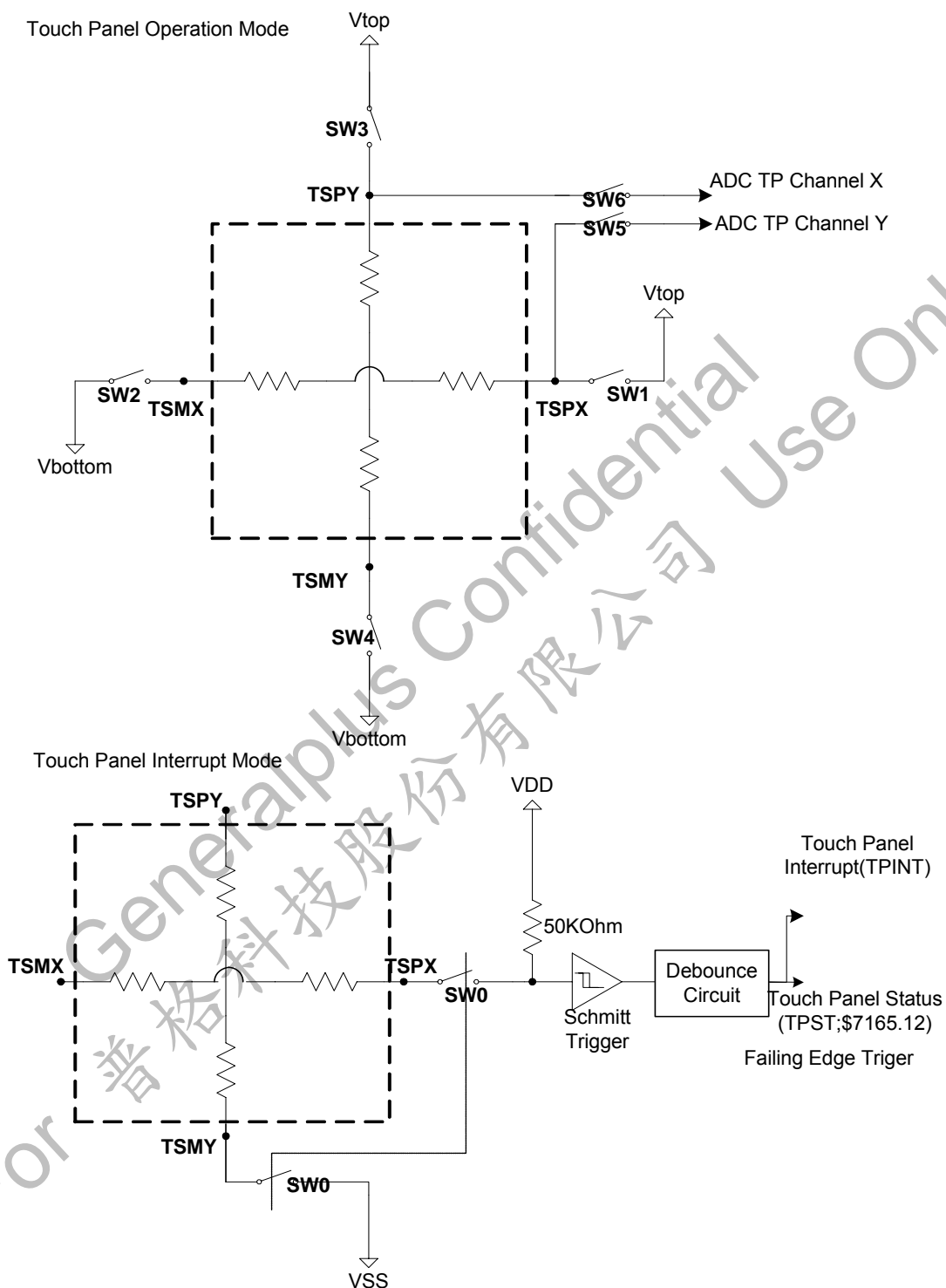
A touch panel is composed of two transparent resistive layers, separated by insulating spacers. We can consider touch panel as two normally-disconnected resistors: a horizontal resistor and a vertical resistor. Following diagram indicates physical view and equivalent circuit of touch panel.



In general, a touch panel equivalent resistor is less than 1000 Ohms. Typically, please refer to the detailed specification of the touch panel. And some parasitic capacitance exists between two transparent ITO conductors.

There are two basic modes for touch panel operation; one is interrupt mode which has lower power consumption and the other is operating mode, used to probe the touch panel coordinate value (x, y). Note that the time interval in operation mode should be as short as possible since power consumption is very large during operating mode.

The following two diagrams indicate equivalent circuitry of interrupt mode and operating mode.



The area in dash line is the touch panel equivalent circuit outside of GPL162002A/162003A. The TSPX and TSMX are connecting pins to touch panel X-Layer (equivalent horizontal resistor). Similarly, TSPY and TSMY are connecting pins to touch panel Y-Layer (equivalent vertical resistor). When stylus taps on touch panel, X-Layer and Y-Layer are inter-connected. In other word, horizontal resistor and vertical resistor can be considered as electrical contact.

The following table depicts switches status in different modes.

	SW0	SW1	SW2	SW3	SW4	SW5	SW6
Interrupt Mode	ON	OFF	OFF	OFF	OFF	OFF	OFF
Operation Mode (X-axis measurement)	OFF	ON	ON	OFF	OFF	OFF	ON
Operation Mode (Y-axis measurement)	OFF	OFF	OFF	ON	ON	ON	OFF

During touch panel interrupt mode and stylus tapping on touch panel, there will be a falling edge signal after Schmitt trigger. This signal informs GPL162002A/162003A that the touch panel is tapped.

During touch panel operation mode and stylus tapping on touch panel, X coordinate value can be determined by measuring the Y layer (from TSPY pin) according to voltage-dividing principle. Similarly, Y coordinate value can be determined by probing the X layer (from TSPX Pin). Note that if stylus does not tap on touch panel, the x or y coordinate value obtained from operation mode will be invalid. Therefore, software must ensure that stylus is tapping while measuring x and y coordinate value. Typically a tap on the touch panel lasts about several tens of micro-seconds (us).

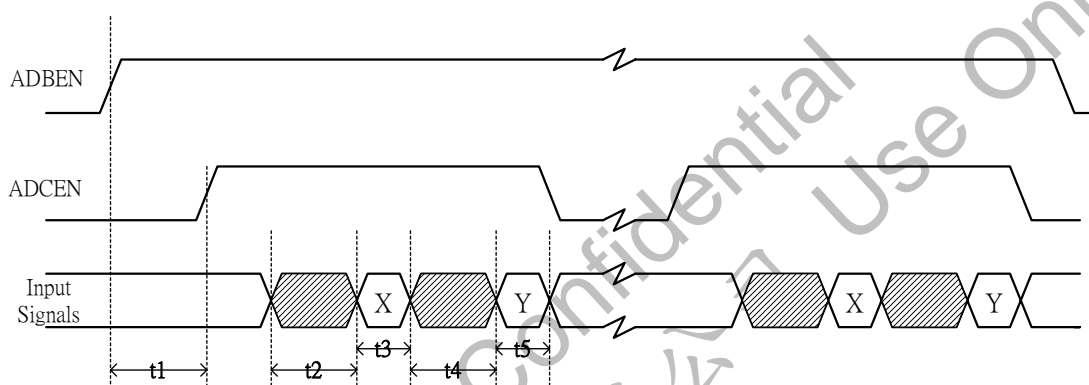
During touch panel interrupt mode, no matter stylus is tapping on the touch panel or not, the system will not consume a huge amount of power because a 50K Ohms pull high resistor exists on TSPX. As a result, programmers should make touch panel in interrupt mode at most of time (even when stylus is tapping on). If the touch panel is tapped and x, y coordinate values are required, turn to the operating mode, and get ADC values. Finally after acquiring ADC values, turn back to interrupt mode immediately.

To check the touch panel tapped or untouched, GPL162002A/162003A provides interrupt and polling mechanisms. After CPU registers are set properly, stylus tap operation can trigger INT and wake GPL162002A/162003A up from halt mode or sleep mode, just like other interrupt or key wakeup events. On the other hand, users can also monitor the stylus tap status by polling. The polling is very important because it is the only way to determine if x and y coordinated values are corrected. The interrupt setup and usage, and wakeup event setup are the same as other interrupt and wakeup events.

Note that there is a de-bounce circuit on stylus tap signal. Therefore, it is not necessary to perform software de-bounce while obtaining touch panel x and y coordinate values.

The time interval between operating mode start and AD conversion start should not be too short. It may take some time to wait that voltage level is stable (RC effect) because a parasitic capacitance may exist on touch panel.

The timing scheme for touch panel coordinate value acquisition is depicted in the following diagram.



t1= ADC BIAS voltage setup time (ADC initialization warm up time)

t2= X measurement switches and input select setup time

t3= X - coordinate value acquisition time (ADC conversion time)

t4= Y measurement switches and input select setup time

t5= Y - coordinate value acquisition time (ADC conversion time)

Following are pseudo codes about touch panel coordinate value acquirement process:

- Step 0: Touch panel is in interrupt mode
- Step 1: Touch panel tapped interrupt event occurs
- Step 2: Turn on ADC
- Step 3: Switch to operation mode measurement X
- Step 4: Delay (for waiting signal stable)
- Step 5: Start AD conversion
- Step 6: Poll AD conversion ready
- Step 7: Obtain X coordinate value by acquiring AD data
- Step 8: Switch to operation mode measurement Y
- Step 9: Delay (for waiting signal stable)
- Step 10: Start AD conversion
- Step 11: Poll AD conversion ready
- Step 12: Obtain Y coordinate value by acquiring AD data

Step 13: Turn off ADC

Step 14: Turn back to touch panel interrupt mode

In the application that power consumption is not a big issue; programmers might not turn ADC on and off frequently as previous description. Instead, Programmers can turn on ADC while system powers on, and turn off ADC when system goes into standby mode. As the programming example provides touch panel Interrupt service routine in this case, programmers can hook this ISR on a 128Hz or 256Hz Interrupt vector.

16.4 Voice Recorder (HQADC operates mode)

GPL162002A/162003A provides a high quality ADC for voice recording. After ADMCLK_EN is 1 and PWADL/PWADR is turned on, the HQADC will start to record the voice depending on the LINEINS settings. The operation current is around 10 ~ 15 mA. The record sample rate is fixed to 48 KHz. If programmers want to decrease the sample rate, use DIV_REC record register to control the hardware FIFO.

Monophonic and stereo recording functions can be selected on GPL162002A/162003A. In monophonic record mode, only left channel will be record. At each AD sample time, one ADC data is generated. In stereo record mode, left and right channels are used. At each AD sample time, there are two ADC data generated. Then, programmers need to get P_ASADC_Data (0x7964) twice, the first data is from left channel and second data is from right channel input.

The procedure to turn on HQADC is: enable ADC clock (0x7807. b4) → enable HQADC clock (0x7970. b0) → setup HQADC control register (0x7970~0x7973) → enable auto-sample mode (0x7960. b7) → start HQADC auto sample.

16.5 Analog Input Control Pin Configuration

SAR ADC interface signals

Name	I/O	Description
TSMY	I	Touch Panel Y-axis Bottom pin (shared with GPIO PortB15)
TSMX	I	Touch Panel X-axis Bottom pin (shared with GPIO PortB14)
TSPY	I	Touch Panel Y-axis Top pin (shared with GPIO PortB13)
TSPX	I	Touch Panel X-axis Top pin (shared with GPIO PortB12)
VADREF*	O	ADC Reference Voltage (supported by GPL162002A/162003A built-in logic)
LINEIN1	I	SAR ADC Input Channel A
LINEIN2	I	SAR ADC Input Channel B
LINEIN3	I	SAR ADC Input Channel C (shared with GPIO PortB10)
LINEIN4	I	SAR ADC Input Channel D (shared with GPIO PortB11)

HQADC interface signals

Name	I/O	Description
MICBIAS	O	Buffered voltage output suitable for electret-microphone-capsule biasing. Voltage level is 3/4 VCCADC
MICIN	I	Microphone input
LINEINR	I	Right channel line input
LINEINL	I	Left channel line input
FMINR	I	Right channel FM input
FMINL	I	Left channel FM input
ADRFLT	O	Right channel anti-aliasing filter capacitor. For ADC
ADLFLT	O	Left channel anti-aliasing filter capacitor. For ADC
TESTP	I/O	Test-mode positive input or output
TESTN	I/O	Test-mode negative input or output

16.6 Control Registers

ADC control register summary table

Name	Address	Description
P_ADC_Setup	0x7960	ADC Setup Register
P_MADC_Ctrl	0x7961	Manual Mode ADC Control Register
P_MADC_Data	0x7962	Manual Mode ADC Data Register
P_ASADC_Ctrl	0x7963	Auto Sample Mode ADC Control Register
P_ASADC_Data	0x7964	Auto Sample Mode ADC Data Register
P_TP_Ctrl	0x7965	Touch Panel Control Register
P_HQADC_Ctrl	0x7970	High Quality ADC control
P_HQADC_PGAS	0x7971	High Quality ADC MICIN pre gain setting.
P_HQADC_RGAIN	0x7972	High Quality ADC LINEINR gain setting.
P_HQADC_LGAIN	0x7973	High Quality ADC LINEINL gain setting.

P_ADC_Setup		0x7960					ADC Setup Register										
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		ADBEN	ADCEN	-	-	-	CLKSEL		ASEN		-	-	-	-	ASMEN		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	ADBEN	R/W	SAR AD Bias Reference Voltage Enable This bit is for SAR ADC only.	0= Disabled 1= Enabled
14	ADCEN	R/W	Enables SAR ADC, for SAR ADC only When ADC is enabled, power consumption will increase rapidly; therefore, turn ADC on only when it is necessary, and turn off ADC immediately after ADC data is successfully obtained. If programmers write 1 to this bit, the ADCADE will turn-on all the time, If this bit is set to 0, the ADCADE will be turn-on only when the manual or auto sample ADC operation is active for reducing the power consumption.	0= ADCADE will be turned-on only when the manual or auto sample ADC operation 1= ADCADE will be turned-on all the time
[13:11]			Reserved	
[10:8]	CLKSEL	R/W	SAR ADC Conversion Time Select In order to increase the sampling speed when system clock is slow. Change this register will reduce the clock cycles needed for an ADC conversion. These bits are for ASR ADC only.	000 = ADC Conversion use 512 SYSCLK 001 = ADC Conversion use 256 SYSCLK 010 = ADC Conversion use 128 SYSCLK

Bit	Function	Type	Description	Condition
				011 = ADC Conversion use 64 SYSCLK 100 = ADC Conversion use 1024 SYSCLK 101 = ADC Conversion use 2048 SYSCLK 110~111: Reserved
7	ASEN	R/W	Auto Sampling Mode Enable, for HQADC only. If this bit is set to “1”, ADC auto sample operation can be applied on HQADC. ADC auto-sample-rate can be selected from overflow frequency of divided from 48KHz (0x7970. b[10:8]).	0 = Disable 1 = Enable
[6:3]			Reserved	
2	ASMEN	W	ADC Auto Sampling Mode Enable, for HQADC only. When this bit is set to “1” (ADC auto sample mode start), AD conversion will start automatically. Then hardware will store the fetched ADC data into 16X16 depth FIFO. On the other hand, If this bit is cleared to “0”, ADC auto sample mode stops.	0= STOP (Disabled) 1= START (Enabled)
[1:0]			Reserved	

P_MADC_Ctrl
0x7961
Manual ModeADC Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ADCRIF/C	ADCRIEN	-	-	-	-	-	-	CNVRDY	STRCNV	-	-	-	-	CHSEL	
Default	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	ADCRIF/C	R/W	AD Conversion Ready Interrupt Flag & Clear This bit is set to “1” by hardware if the AD conversion is ready and ADC data is reliable.	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
14	ADCRIEN	R/W	AD Conversion Ready Interrupt Enable If this bit is set to “1”, and AD conversion is ready, hardware will issue an IRQ1 or FIQ to CPU. To select between IRQ1 and FIQ, please refer to Chapter Interrupt .	0= Disabled 1= Enabled
[13:8]			Reserved	

Bit	Function	Type	Description	Condition
7	CNVRDY	R	AD Conversion Ready Indicate bit	0= Not ready, AD data not effect 1= Ready, ADC data is effect
6	STRCNV	W	Manual Start AD Conversion Writing this bit to "1" will start the operation of AD conversion.	0= No Effect 1= START
[5:3]			Reserved	
[2:0]	CHSEL	R/W	Current ADC Channel Selection	000= Selects TP X-axis 001= Selects TP Y-axis 010= Selects LINEIN1 011= Selects LINEIN2 100= Selects LINEIN3 101= Selects LINEIN4 110~111= Reserved

P_MADC_Data 0x7962 Manual ADC Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	MADCDATA												-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:4]	ADCDATA	R	Manual AD Conversion Data After Control bit CNVRDY is set, ADC data will be valid on this register.	
[3:0]			Reserved	

P_ASADC_Ctrl 0x7963 Auto Sample control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ASIF/C	ASIEN	ASFF	ASFOV	DMA	OVER	ASFIL					FIFOLEV				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	ASIF/C	R/W	Auto Sample Mode FIFO Full interrupt flag This bit is set to "1" by hardware if auto sample mode is enabled, and the Auto Sample FIFO Full (including microphone channel) interrupt is asserted. Note that this flag should be clear	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
			<p>after acquiring ADC data from auto sample FIFO. Programmers cannot clear this bit BEFORE acquiring ADC data from the FIFO.</p> <p>If DMA is set to 1, this bit will be clear as long as the data in the FIFO is lower than the trigger level.</p>	
14	ASIEN	R/W	<p>Auto Sample Mode FIFO Full Interrupt Enable</p> <p>This bit is used to enable FIFO full interrupt when Auto Sample mode is activated. If this bit is set to "1", and the FIFO is full (definition of full is depending on ASFIL control bits, P_ASADC_Ctrl.bit [9...5]), hardware will issue an IRQ1 or FIQ to CPU. To select between IRQ1 and FIQ, please refer to Chapter Interrupt.</p>	<p>0= Disabled</p> <p>1= Enabled</p>
13	ASFF	R	<p>Auto Sample Mode FIFO Full Flag</p> <p>This read-only flag is set to "1" by hardware when auto sample FIFO is full.</p>	<p>0= Not Full</p> <p>1= Full</p>
12	ASFOV	R	<p>Auto Sample Mode FIFO Overflow Flag</p> <p>This read-only flag is set to "1" by hardware if auto sample FIFO has been full and last data has been replaced by the latest sampled ADC data.</p>	<p>0= Auto Sample FIFO is not overflow</p> <p>1= Auto Sample FIFO is overflow</p>
11	DMA	R/W	<p>DMA mode</p> <p>The ADC auto sample mode is connected to the DMA channel source 8, the DMA transfer will not clear ASIF. After DMA mode is enabled, ASIF will be cleared automatically when FIFO level is lower than designated level.</p>	<p>0 = Interrupt mode, programmer needs to write 1 to ASIF to clear the ASIF.</p> <p>1 = DMA mode, programmer do not need to write 1 to ASIF, the ASIF will be clear automatically when FIFO level is lower than trigger level.</p>
10	OVER	R/W	<p>Auto Sample FIFO Over Write Mode</p> <p>This register is to decide to overwrite or to skip incoming data right after auto sample FIFO is full.</p>	<p>0 = The further write to the full FIFO will be skipped.</p> <p>1 = The further write to the full FIFO will overwrite the last written data in the FIFO.</p>

Bit	Function	Type	Description	Condition
[9:5]	ASFIL	R/W	Auto Sample Mode FIFO Full Interrupt Level The control bits are used to setup FIFO full interrupt timing. It defines the number of data left in FIFO to be considered as full by hardware. The smaller the value is, the more often the FIFO empty interrupt occurs. The larger the value is, the less frequent the FIFO full interrupt happens. Consequently, it saves CPU bandwidth.	FIFO Full Interrupt issue timing 00000= when data no. in FIFO >=0 00001= when data no. in FIFO >=1 00010= when data no. in FIFO >=2 00011= when data no. in FIFO >=3 00100= when data no. in FIFO >=4 00101= when data no. in FIFO >=5 00110= when data no. in FIFO >=6 00111= when data no. in FIFO >=7 01000= when data no. in FIFO >=8 01001= when data no. in FIFO >=9 01010= when data no. in FIFO >=10 01011= when data no. in FIFO >=11 01100= when data no. in FIFO >=12 01101= when data no. in FIFO >=13 01110= when data no. in FIFO >=14 01111= when data no. in FIFO >=15 10000= when data no. in FIFO =16 10001~11111= Reserved
[3:0]	FIFOLEV	R	Auto Sample Mode FIFO Full Interrupt Level This read-only flag reports the number of ADC sampled data in the 16X16-bit ring type FIFO.	00000= 0 data is in FIFO 00001= 1 data is in FIFO 00010= 2 data is in FIFO 00011= 3 data is in FIFO 00100= 4 data is in FIFO 00101= 5 data is in FIFO 00110= 6 data is in FIFO 00111= 7 data is in FIFO 01000= 8 data is in FIFO 01001= 9 data is in FIFO 01010= 10 data is in FIFO 01011= 11 data is in FIFO 01100= 12 data is in FIFO 01101= 13 data is in FIFO 01110= 14 data is in FIFO 01111= 15 data is in FIFO 10000= 16 data is in FIFO 10001~11111= Reserved

P_ASADC_Data
0x7964
Auto Sample Data register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ASADC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ASADC	R	Auto Sample Mode FIFO Data If auto sample mode FIFO is not empty, programmers can obtain ADC data from this control register.	

P_TP_Ctrl
0x7965
Touch Panel Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TPIF/C	TPIEN	TPEN	TPST	TMOD	-	-	-	-	-	-	DBEN	-	-	DBTSEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	TPIF/C	R/W	Touch panel interrupt flag This bit is set to "1" by hardware if touch panel interrupt is asserted. (Touch panel is at interrupt mode, and a stylus is tapped on touch screen.)	Read 0= Not Occurred Read 1= Occurred Write 0= No Effect Write 1= Clear the flag
14	TPIEN	R/W	Touch Panel Interrupt Enable If this bit is set to "1", and at the time when stylus is tapped on touch screen, this hardware will issue an IRQ1 or FIQ to CPU. To select between IRQ0 and FIQ, please refer to Chapter Interrupt .	0= Disabled 1= Enabled
13	TPEN	R/W	Touch Panel Interface enable When this bit is set to "1", PortB [15:12] becomes Touch Panel Interface. These I/O pins cannot be used as GPIO function.	0= Disabled 1= Enabled
12	TPST	R	Touch panel stylus tapped status This bit is valid only when touch panel is at interrupt mode.	1= touch panel stylus tapped 0= touch panel stylus not tapped
11	TMOD	R/W	Touch Panel Mode Refer to Section: Touch Panel Interface for detailed operation schemes.	0 = Interrupt Mode 1= Operation Mode Note: Generalplus suggests programmers set TSPX to high and wait a minute for stable when first time enabling touch panel function and then setting to interrupt mode. This method is to avoid dummy TP interrupt occur at first time when touch panel is enabled and set to interrupt mode.

Bit	Function	Type	Description	Condition
[10:5]			Reserved	
4	DBEN	R/W	De-bounce enable Note: Before CPU goes to wait, halt or sleep mode, this control bit must be always set to "1" once the touch panel interface is enabled (TPEN=1). Otherwise, touch panel operation might be failed sometime.	0= Disabled 1= Enabled
[3:2]			Reserved	
[1:0]	DBTSEL	R/W	De-bounce timing select (default=00)	00= de-bounce by SYSCLK/1024 01= de-bounce by SYSCLK/2048 10= de-bounce by SYSCLK/4096 11= de-bounce by SYSCLK/8192

P_HQADC_Ctrl
0x7970
High Quality ADC control

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	ADVOR	ENHP	DIV_REC	MONO	BOOST	INMODE	PWADL	PWADR	MICBIAS	ADMCLK				
Default	0	0	0	0	1	0	0	0	0	0	1	1	1	1	1	0

Bit	Function	Type	Description	Condition
[15:14]			Reserved	
[13:12]	ADVOR	R/W	HQADC input limit range select	00: 0.84 * full range 01: 0.71 * full range 10: 0.60 * full range 11: 0.50 * full range
11	ENHP	R/W	HQADC High Pass filter enable register When HQADC recording is used, this bit need be set to 1.	0: Disable 1: Enable
[10:8]	DIV_REC	R/W	Record Sample Rate Control Register After the initialization of the HQADC, the HQADC will output recorded data in 48KHz sampling rate. When users do not need to sample data in such high sampling rate, this register can be used for reducing the sample rate of recording. This is done by skipping the recorded data we don't need.	000= Record in 48KHz sample rate 001= Record in 24KHz sample rate 010= Record in 16KHz sample rate 011= Record in 12KHz sample rate 100= Record in 9.6KHz sample rate 101= Record in 8KHz sample rate 110= Record in 6.9KHz sample rate 111= Record in 6KHz sample rate
7	MONO	R/W	Mono Record Control Register This bit is to choose the monophonic	0= Stereo mode 1= Mono mode, only left channel will

Bit	Function	Type	Description	Condition
			or stereo mode of recording. If stereo mode is selected, then at each sampling time, the left and right channel data will be sent to FIFO. If mono mode is selected, then at each sampling time, only left channel data will be sent to FIFO.	be recorded.
6	BOOST	R/W	Internal Boost Amplifier Control	0= Disable boost amplifier 1= Enable boost amplifier
[5:4]	INMODE	R/W	HQADC Input Source Select There are three input sources on GPL162002A/162003A: MIC, LINE, and FM. These bits are to select the input source.	00: MIC-in 01: Line-in 10: FM-in 11: All off
3	PWADL	R/W	HQADC Left Channel ADC Power Control	0= Power on 1= Power down
2	PWADR	R/W	HQADC Right Channel ADC Power Control	0= Power on 1= Power down
1	MICBIAS	R/W	HQADC Microphone Bias-Voltage Output Power Control This bit is useful only when INMODE is selected to MIC channel.	0= Bias-voltage power on 1= bias-voltage power down
0	ADMCLK	R/W	HQADC Main Clock Enable The HQADC needs a 12.288MHz clock for the digital filter. Before enabling the HQADC, programmers must first enable DAPLL (0x7807.b4) and wait PLL stable, and then set this bit to 1 to initiate HQADC clock.	0= Disable 1= Enable

P_HQADC_PGAS
0x7971
High Quality ADC MICIN pre gain setting

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	PGAS				
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

Bit	Function	Type	Description	Condition
[15:5]			Reserved	
[4:0]	PGAS	R/W	The gain setting of MICIN PGA.	00000: 33 dB 00001: 31.5 dB 00010: 30 dB 00011: 28.5 dB 00100: 27 dB

Bit	Function	Type	Description	Condition
				<p>..... decrease 1.5dB on each level</p> <p>11101: -10.5 dB</p> <p>11110: -12 dB</p> <p>11111: -∞ dB (mute)</p>

P_HQADC_RGAIN
0x7972
High Quality ADC LINEINR gain setting

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ADROVP	ADROVN	ADROV_IEN	-	-	-	-	-	-	-	-	LINEGR				
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
15	ADROVP	R/W	<p>HQADC Right Channel Line-in Top Overflow interrupt flag</p> <p>This bit is set to 1 by hardware if HQADC right channel line-in top overflow occurs. Programmers can use this register to determine the setting of LINEGR. When top overflow occurs, programmers need to reduce LINEGR.</p>	<p>Read 0= Not Occurred</p> <p>Read 1= Occurred</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>
14	ADROVN	R/W	<p>HQADC Right Channel Line-in Bottom Overflow interrupt flag</p> <p>This bit is set to 1 by hardware if HQADC right channel line-in bottom overflow occurs. Programmers can use this register to determine the setting of LINEGR. When bottom overflow occurs, programmers need to reduce LINEGR.</p>	<p>Read 0= Not Occurred</p> <p>Read 1= Occurred</p> <p>Write 0= No effect</p> <p>Write 1= Clear the flag</p>
13	ADROV_IEN	R/W	<p>HQADC Right Channel Overflow Interrupt Enable</p> <p>If this bit is set to 1, and at the time when ADROVP or ADVRON is set to 1, hardware will issue an IRQ1 or FIQ to CPU.</p> <p>To select between IRQ1 and FIQ, please refer to Chapter Interrupt.</p>	<p>0= Disable</p> <p>1= Enable</p>
[12:5]			Reserved	
[4:0]	LINEGR	R/W	The gain setting of Right Channel Line-in.	<p>00000: 12 dB</p> <p>00001: 10.5 dB</p> <p>00010: 9 dB</p> <p>00011: 7.5 dB</p> <p>00100: 6 dB</p> <p>..... decrease 1.5dB on each</p>

Bit	Function	Type	Description	Condition
				level 11101: -31.5 dB 11110: -33 dB 11111: -∞ dB (mute)

P_HQADC_LGAIN
0x7973
High Quality ADC LINEINL gain setting

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ADLOVP	ADLOVN	ADLOV_IEN	-	-	-	-	-	-	-	-	LINEGL				
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Function	Type	Description	Condition
15	ADLOVP	R/W	HQADC Left Channel Line-in or MICIN Top overflow interrupt flag This bit is set to 1 by hardware if HQADC left channel line-in or MICIN overflow occurs. Programmers can use this register to determine the setting of LINEGL or PGAS. When top overflow occurs, programmers need to reduce LINEGL or PGAS.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	ADLOVN	R/W	HQADC Left Channel Line-in or MICIN Bottom overflow interrupt flag This bit is set to 1 by hardware if HQADC left channel line-in or MICIN bottom overflow occurs. Programmers can use this register to determine the setting of LINEGL or PGAS. When bottom overflow occurs, programmers need to reduce LINEGL or PGAS.	Read 0= Not Occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
13	ADLOV_IEN	R/W	HQADC Left Channel Overflow Interrupt Enable If this bit is set to 1, and at the time when ADLOV is set to 1, this hardware will issue an IRQ1 or FIQ to CPU. To select between IRQ1 and FIQ, please refer to Chapter Interrupt .	0= Disable 1= Enable
[12:5]			Reserved	
[4:0]	LINEGL	R/W	The gain setting of Left Channel Line-in.	00000: 12 dB 00001: 10.5 dB 00010: 9 dB 00011: 7.5 dB 00100: 6 dB

Bit	Function	Type	Description	Condition
			 decrease 1.5dB on each level 11101: -31.5 dB 11110: -33 dB 11111: -∞ dB (mute)

16.7 Program Example

Voice Record on Microphone channel & Touch panel

```

Int off
r1 = 0x0dc1                                // Set 8KHz sampling rate, enable AD clock,
[P_HQADC_Ctrl] = r1                        //stereo mode, R-L channel power on

R1=0xc100                                // enable auto-sample interrupt, FIFO level 8
[P_ASADC_Ctrl]=r1

R1=[ P_ADC_Setup]                        //enable HQADC, enable SAR ADC
R1|=0xc080
[P_ADC_Setup]=r1
r1=0xc000
[P_MADC_Ctrl]=r1                        //enable SAR AD vonversion ready interrupt
r1=0x2810                                //enable touch panel interface, operation //mode,
[P_TP_Ctrl]=r1                            X channel
r1 = 0x0080
[P_INT_Priority1] = r1

r2 = 5
ds = r2
r3 = 0x0000

call    F_DelayADCStable                // Wait for ADC bias voltage stable

fiq on
irq on
r1 = [P_ADC_Setup]
r1 = r1 | 0x0004                        // Start Auto Sample Operation
[P_ADC_Setup] = r1

r1=[ P_MADC_Ctrl]
r1|=0x040                                //start SAR AD converter
[P_MADC_Ctrl]=r1

call    K_WaitKeyTrigger                // Wait for key trigger to stop voice recoding

```




© Generalplus Technology Inc.

Touch Panel Interface Example (X channel only)

```
F_TouchPanel_ISR:      .PROC
                        push r1 to [sp]
                        r1 = [P_INT_Status1]
                        r1 = r1 & 0x4000
                        jz    L_EndTPSample?           // If AD convert ready
                        r1=[ P_MADC_Data]              //get manual AD data
                        [R_Xvalue]=r1                  //store X value
                        r1=[ P_MADC_Ctrl]
L_EndTPSample?         [P_MADC_Ctrl]=r1               // clear int flag
                        pop r1 from [sp]
                        retf;
                        .ENDP;
```

L_TPNNotTapped?:

Note that Generalplus also provides complete source codes of touch panel and voice recording. Programmers might contact Generalplus for corresponding executable files.

17 NAND Flash Interface

17.1 Introduction

As a mass storage device, NAND-gate flash becomes more and more popular nowadays for its large capacity and relatively low price. To extend GPL162002A/162003A's application field, a NAND flash interface is incorporated. GPL162002A/162003A provides easy-to-use control registers to generate read/write signals to access NAND flash. In addition, a DMA channel is also provided to speed up NAND Flash data access. GPL162002A/162003A supports 8-bit or 16-bit NAND type flash data access along with hardware ECC (Error Correction Code) and checksum.

- Support the an interface to access 8-bit or 16-bit NAND flash memory.
- Support hardware ECC & checksum.
- Programmable setup/hold timing for accessing.

17.2 NAND Flash Control Pin Configuration

Name	I/O	Description
NFWEB	O	Write Enable (shared with GPIO PortB5)
NFOEB	O	Read Enable (shared with GPIO PortB6)
NFCLE	O	Command Latch Enable (shared with GPIO PortB7)
NFALE	O	Address Latch Enable (shared with GPIO PortB8)
NFRDY	I	Ready/Busy Output (shared with GPIO PortB9)

17.3 Control Register

Nand Flash Control Register Summary Table

Name	Address	Description
P_NF_Ctrl	0x7850	NAND Flash Control Register
P_NF_CMD	0x7851	NAND Flash Command Register
P_NF_AddrL	0x7852	NAND Flash Low Address Register
P_NF_AddrH	0x7853	NAND Flash High Address Register
P_NF_Data	0x7854	NAND Flash Data Register
P_NF_INT_Ctrl	0x7855	NAND Flash DMA / INT Control Register
P_ECC_Ctrl	0x7857	ECC Control Register
P_ECC_LPRL_LB	0x7858	ECC Low Byte Line Parity LSB Register
P_ECC_LPRH_LB	0x7859	ECC Low Byte Line Parity MSB Register
P_ECC_CPR_LB	0x785A	ECC Low Byte Column Parity Check LSB Register
P_ECC_LPR_CKL_LB	0x785B	ECC Low Byte Line Parity Check LSB Register
P_ECC_LPR_CKH_LB	0x785C	ECC Low Byte Line Parity Check MSB Register

Name	Address	Description
P_ECC_CPCKR_LB	0x785D	ECC Low Byte Column Parity Check Register
P_ECC_ERR0_LB	0x785E	ECC Low Byte Error Flag0
P_ECC_ERR1_LB	0x785F	ECC Low Byte Error Flag1
P_ECC_LPRL_HB	0x7848	ECC High Byte Line Parity LSB Register
P_ECC_LPRH_HB	0x7849	ECC High Byte Line Parity MSB Register
P_ECC_CPR_HB	0x784A	ECC High Byte Column Parity Register
P_ECC_LPR_CKL_HB	0x784B	ECC High Byte Line Parity Check LSB Register
P_ECC_LPR_CKH_HB	0x784C	ECC High Byte Line Parity Check MSB Register
P_ECC_CPCKR_HB	0x784D	ECC High Byte Column Parity Check Register
P_ECC_ERR0_HB	0x784E	ECC High Byte Error Flag0
P_ECC_ERR1_HB	0x784F	ECC High Byte Error Flag1
P_CHECKSUM0_LB	0x7830	NAND Flash Low Byte Check Sum Low Value
P_CHECKSUM1_LB	0x7831	NAND Flash Low Byte Check Sum High Value
P_CHECKSUM0_HB	0x7832	NAND Flash High Byte Check Sum Low Value
P_CHECKSUM1_HB	0x7833	NAND Flash High Byte Check Sum High Value

P_NF_Ctrl
0x7850
NAND Flash Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFBF	8or16	-	-	-	-	-	-	NFC7	NFC6	NFC5	NFC4	NFC3	NFC2	NFC1	NFC0
Init	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
15	NFBF	R	Read back busy status (RB_n)	1: ready 0: busy
14	NF8or16	R/W	8-bit or 16-bit Nand flash memory data access.	0: 8 bits 1: 16 bits
[13:8]			Reserved	
[7:0]	NFCTRL	R/W	Adjustable setup/hold time tREH=(NFCTRL[7:6] + 1) x CPUCLK tREA=(NFCTRL[5:4] + 1) x CPUCLK tWH=(NFCTRL[3:2] + 1) x CPUCLK tWP=(NFCTRL[1:0] + 1) x CPUCLK	[7:6]: tREH, RE_n high pulse [5:4]: tREA, Access time [3:2]: tWH, WE_n high pulse [1:0]: tWP, WE_n low pulse

P_NF_CMD
0x7851
NAND Flash Command Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFCMD															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFCMD	W	Write Command instruction. Write "Command value" to the register, and NAND FLASH interface will write this "Command value" to NAND FLASH memory automatically.	[15:0]: Command value For example: 00H: Read A area 01H: Read B area 50H: Read C area 80H, 10H: Page program 70H: read Status 90H: read ID FFH: reset 60H,D0H: Block erase

P_NF_AddrL
0x7852
NAND Flash ADDR Low Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	2 nd Cycle								1 st Cycle							
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFADDRL	W	Write Low Word Address instruction Write 1 st cycle and 2 nd cycle "Address value" to the register, and then NAND FLASH interface will write this Address value to NAND FLASH memory automatically.	[15:0]: Address Value

P_NF_AddrH
0x7853
NAND Flash ADDR High Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	4 th Cycle								3 rd Cycle							
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFADDRH	W	Write High Word Address instruction Write 3 rd cycle and 4 th cycle of "Address value" to the register, and then NAND FLASH interface will write this Address value to NAND FLASH memory.	[15:0]: Address Value

Some of Nand Flash commands only take two address cycle, such as Block erase; however, programmers still need to write dummy values to P_NF_AddrH register. Otherwise, the address will not be sent to the Nand Flash.

P_NF_Data		0x7854								NAND Flash Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		NFDATA															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NFDATA	R/W	Read/Write Data instruction Write "Data value" to the register, and NAND FLASH interface will write this Data value to NAND Flash memory automatically. A Read operation will read Data value from NAND Flash memory.	[15:0]: Data Value, 16-bit type Nand Flash [7:0]: Data Value, 8-bit type Nand Flash

When the Nand Flash is 8-bit type, the data in P_NF_Data [15:8] register is invalid. The data value written to / read from the Nand Flash is effective only in the lower byte of the P_NF_Data register. When the Nand Flash is 16-bit type, the data in P_NF_Data is all valid.

P_NF_INT_Ctrl		0x7855							DMA/INT Control Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function		REQF/C	DMAEN	INTEN	-	ADR4EN	ADR3EN	ADR2EN	-	-	-	-	-	-	-	-	-	
Init		0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	REQF/C	R/W	NAND Flash Access Request Flag. Write "1" to clear the flag. This bit is set to "1" by hardware after the CLE and ALE commands. When NAND flash is ready to Read/Write, the Request flag is asserted, Clear it to "0" after Reading/Writing NAND Flash memory through port 0x7854.	Read 0= Not Occurred Read 1= Occurred Write 0 = No effect Write 1 = Clear the flag
14	DMAEN	R/W	NAND Flash DMA Access Enable. If this bit is set to "1", and if NAND Flash Access Request occurs, hardware will issue a DMA request to DMA controller. If this bit is cleared to "0", this request will be masked. When Nand Flash DMA is enabled, programmers should set DMA mode as software mode or external demand mode. Please refer to Chapter DMA for details.	0= Disable 1= Enable
[13]	INTEN	R/W	NAND Flash Access Interrupt Enable. If this bit is set to "1", and if NAND Flash Access Request occurs, hardware will issue an IRQ5 or	0= Disable 1= Enable

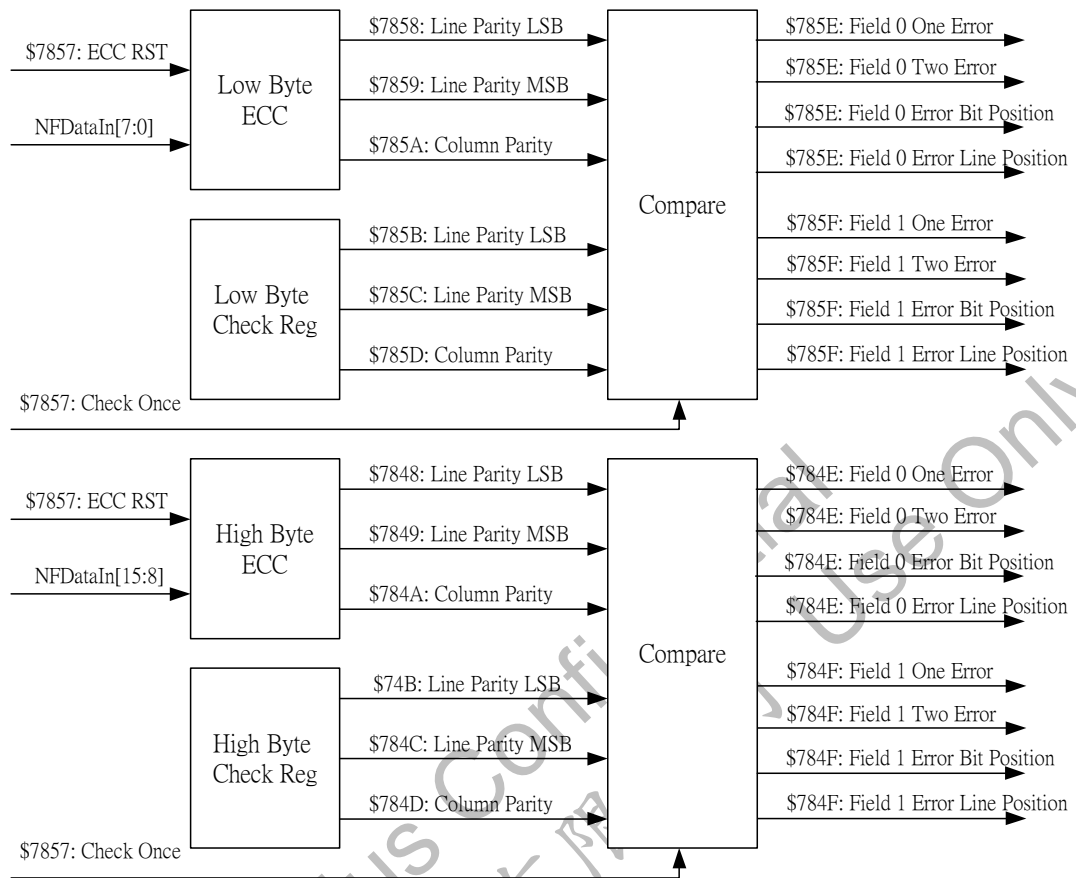
Bit	Function	Type	Description	Condition
			FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ5 and FIQ, please refer to Chapter Interrupt .	
12			Reserved	
11	ADR4EN	R/W	NAND Flash memory fourth cycle A25~A32 address enable. If NAND Flash uses 32-bit address, The ADR4EN and ADR3EN both need to be set to 1.	0: disable 1: enable
10	ADR3EN	R/W	NAND Flash memory third cycle A17~A24 address enable.	0: disable 1: enable
9	ARD2EN	R/W	NAND Flash memory second cycle A16~A8 address enable.	0: disable 1: enable
[8:0]			Reserved	

17.4 NAND Flash ECC & Checksum

GPL162002A/162003A internal ECC & checksum calculation logic can be controlled by software via P_ECC_Ctrl.bit2. When it is set to "0", it will calculate read / write data appears on the NAND flash data bus. The value after ECC will be output to the ECC logic, P_ECC_LPRL_LB, P_ECC_LPRH_LB, P_ECC_CPR_LB, P_ECC_LPRL_HB, P_ECC_LPRH_HB, and P_ECC_CPR_HB in every calculation. And value after checksum calculation will be output to P_CHECKSUM0_LB, P_CHECKSUM1_LB, P_CHECKSUM0_HB, and P_CHECKSUM1_HB in every calculation.

In addition, because of the nature of ECC and checksum logic, only after exactly 512 bytes (256 words) in 8-bit type Nand flash and 256 words (512 bytes) in 16-bit type Nand flash of data are given to the ECC and check sum logic, the calculation result is valid. To guarantee a correct ECC result, users must reset P_ECC_Ctrl.bit0 by filling 1 at the beginning of reading / writing a page. When the NAND flash connected to GPL162002A/162003A is 8-bit type, only the registers 0x7858 ~ 0x785F are valid in calculating ECC and only 0x7830~0x7831 are valid in calculating checksum. When the NAND flash connected to GPL162002A/162003A is 16-bit type, all the ECC and checksum registers are valid.

In checksum function, GPL162002A/162003A does not support the compare function, but it does have that in ECC function.


P_ECC_Ctrl
0x7857
ECC Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	ECCSPT	CKP	ERST
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:3]			Reserved	
2	ECCSPT	W	ECC Stop Calculation Write "1" to this bit will stop ECC and checksum calculation, all parity registers will keep the previous value.	1: Stop calculate 0: Calculate
1	CKP	W	Write "1" to check parity (Line or Column) once. The Low Byte Error Flag will be shown on 0x785E and 0x785F, and the High Byte Error Flag will be shown on 0x784E and 0x784F. In checksum function it does not support check function.	1: Check once 0: no action
0	ERST	W	Reset ECC. The reset action must be done before any data is transferred to the ECC and checksum module.	1: Reset 0: not Reset

P_ECC_LPRL_LB		0x7858				ECC Low Byte Line parity LSB Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		LPRL															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRL	R	The ECC Line parity register (LSB)	

P_ECC_LPRH_LB		0x7859				ECC Low Byte Line parity MSB Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	LPRH																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:0]	LPRH	R	The ECC Line parity register (MSB)	

P_ECC_CPR_LB		0x785A				ECC Low Byte Column parity Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-			-			-			-			CPR				
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPR	R	The ECC Column parity register	

All 16-bit value of P_ECC_LPRL_LB, and P_ECC_LPRH_LB, and all 12-bit value of P_ECC_CPR_LB are valid. As a result, when the Nand Flash is chosen as 8-bit type, it needs to write low byte and high byte to P_NE_Data, respectively.

P_ECC_LPR_CKL_LB				0x785B				ECC Low Byte Line parity Check LSB Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRCKL	R/W	The ECC Line parity Check register (LSB).	

P_ECC_LPR_CKH_LB				0x785C				ECC Low Byte Line parity Check MSB Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	LPRCKH																		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description	Condition
[15:0]	LPRCKH	R/W	The ECC Line parity Check register(MSB)	

P_ECC_CPCKR_LB 0x785D
ECC Low Byte Column parity Check Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	CPRCK											
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPRCK	R/W	The ECC Column parity Check register	

P_ECC_ERR0_LB 0x785E
ECC Low Byte Field-0 Error Flag

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE							
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:13]			Reserved	
[12]	2ERR	R	There are two error bits.	0: error free 1: error
[11]	1ERR	R	There is one error bit.	0: error free 1: error
[10:8]	FAILBIT	R	The error bit position.	3: error free others(m): error on bit m
[7:0]	FAILLINE	R	The error line position.	255: error free others(n): error on line n

P_ECC_ERR1_LB 0x785F
ECC Low Byte Field-1 Error Flag

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE							
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:13]			Reserved	
[12]	2ERR	R	There are two error bits.	0: error free 1: error
[11]	1ERR	R	There is one error bit.	0: error free 1: error
[10:8]	FAILBIT	R	The error bit position.	3: error free others(m): error on bit m

Bit	Function	Type	Description	Condition
[7:0]	FAILLINE	R	The error line position.	255: error free others(n): error on line n

The control register, P_ECC_ERR0_LB, stores the error information of 0~255 bytes of Nand Flash. Moreover, P_ECC_ERR1_LB stores the error information of 256~511 bytes of Nand Flash.

P_ECC_LPRL_HB 0x7848 ECC High Byte Line parity LSB Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRL	R	The ECC Line parity register(LSB)	

P_ECC_LPRH_HB 0x7849 ECC High Byte Line parity MSB Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRH	R	The ECC Line parity register(MSB)	

P_ECC_CPR_HB 0x784A ECC High Byte Column parity Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPR	R	The ECC Column parity register	

P_ECC_LPR_CKL_HB 0x784B ECC High Byte Line parity Check LSB Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRCKL	R/W	The ECC Line parity Check register (LSB)	

P_ECC_LPR_CKH_HB 0x784C
ECC High Byte Line parity Check MSB Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	LPRCKH	R/W	The ECC Line parity Check register(MSB)	

P_ECC_CPCR_HB 0x784D
ECC High Byte Column parity Check Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	CPRCK											
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	CPRCK	R/W	The ECC Column parity Check register	

P_ECC_ERR0_HB 0x784E
ECC High Byte Field-0 Error Flag

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	2ERR	1ERR	FAILBIT				FAILLINE						
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:13]			Reserved	
[12]	2ERR	R	There are two error bits.	0: error free 1: error
[11]	1ERR	R	There is one error bit.	0: error free 1: error
[10:8]	FAILBIT	R	The error bit position.	3: error free others(m): error on bit m
[7:0]	FAILLINE	R	The error line position.	255: error free others(n): error on line n

P_ECC_ERR1_HB 0x784F
ECC High Byte Field-1 Error Flag

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	2ERR	1ERR	FAILBIT				FAILLINE						
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:13]			Reserved	

Bit	Function	Type	Description	Condition
[12]	2ERR	R	There are two error bits.	0: error free 1: error
[11]	1ERR	R	There is one error bit.	0: error free 1: error
[10:8]	FAILBIT	R	The error bit position.	3: error free others(m): error on bit m
[7:0]	FAILLINE	R	The error line position.	255: error free others(n): error on line n

The control register, P_ECC_ERR0_HB, stores the error information of 0~255 high bytes of Nand Flash. Moreover, P_ECC_ERR1_HB stores the error information of 256~511 high bytes of Nand Flash. For more information about ECC check error registers, please refer to **NAND Flash ECC & CheckSum** special note.

P_CHECKSUM0_LB						NAND Flash Low Byte Check Sum Low Value										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM0_LB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHECKSUM0_LB	R/W	NAND Flash low byte 0~255 bytes check sum value.	

P_CHECKSUM1_LB						0x7831		NAND Flash Low Byte Check Sum High Value									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	CHECKSUM1_LB																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:0]	CHECKSUM1_LB	R/W	NAND Flash low byte 256~511 bytes check sum value.	

P_CHECKSUM0_HB				0x7832				NAND Flash High Byte Check Sum Low Value								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM0_HB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHECKSUM0_HB	R/W	NAND Flash High byte 0~255 bytes check sum value. This register is valid only when NAND flash is set to 16-bit mode.	

P_CHECKSUM1_HB				0x7833				NAND Flash High Byte Check Sum High Value								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM1_HB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	CHECKS UM1_HB	R/W	NAND Flash High byte 256~511 bytes check sum value. This register is valid only when NAND flash set to 16-bit mode.	

17.5 Special Note

As for ECC check error registers, here are some descriptions.

For 8-bit type nand flash, the unit of ECC logic calculation is 512 bytes, and the results can correct 1 bit error for each 256 bytes and determine two bit errors for each 256 bytes. For 0~255 byte, the error information is stored in 0x785E (P_ECC_ERR0_LB). For 256~511 byte, the error information is stored in 0x785F (P_ECC_ERR1_LB).

For 16-bit type NAND flash, the unit of ECC logic calculation is 256 words or 512 words.

For 256 words, the ECC result can correct 1 bit for low 256 bytes and high 256 bytes, respectively, and determine two bit errors for low 256 bytes and high 256 bytes, respectively. For 0~255 low byte, the error information is stored in 0x785E (P_ECC_ERR0_LB). For 0~255 high byte, the error information is stored in 0x784E (P_ECC_ERR0_HB).

For 512 words, the error information is described as follows. For 0~255 low byte, the error information is stored in 0x785E (P_ECC_ERR0_LB). For 256~511 low byte, the error information is stored in 0x785F (P_ECC_ERR1_LB). For 0~255 high byte, the error information is stored in 0x784E (P_ECC_ERR0_HB). For 256~511 high byte, the error information is stored in 0x784F (P_ECC_ERR1_HB).

17.6 Program Example

8-bit type Nand Flash with DMA mode

F_WriteNAND_Byte:

```
r1=0x00aa //nand flash initial, byte mode
[P_NF_Ctrl]=r1

r1=0x9000
[P_NF_INT_Ctrl]=r1

r1 = 0x00
[R_DMAFlag] = r1
irq on

r1=0x0200 //dma Channel 0 control reset
[P_DMA_Ctrl0] = r1

r1 = Data_Addr_Low //Source DataAddr low
[P_DMA_SRC_AddrL0] = r1
r1 = Data_Addr_High //Source DataAddr high
[P_DMA_SRC_AddrH0] = r1

r1 = P_NF_Data //Destination address: nand flash data
[P_DMA_TAR_AddrL0] = r1 //port
r1 = 0x00
[P_DMA_TAR_AddrH0] = r1

r1 = 512
[P_DMA_TCountL0] = r1 // Write 512 bytes consecutively
r1 = 0 // Data will first be written to "A" area
[P_DMA_TCountH0] = r1 and
// then "B" area

r1 = [P_DMA_SS]
r1 &= ~0x0f
r1 |= 0x05 //set dma0 source 5(nand flash)
[P_DMA_SS] = r1

r1 = 0xa148
[P_DMA_Ctrl0] = r1

r1 = 0xd400
[P_NF_INT_Ctrl] = r1

//nand flash DMA enable, addr3
```

```

r3 = 0x00                                enable
[P_NF_CMD] = r3

r3 = 0x80                                //read block A area 00H command
[P_NF_CMD] = r3
r1 = 0x00
[P_NF_AddrL] = r1                        //send 80H command
r1 = 0x00
[P_NF_AddrH] = r1                        //write PageAddr low

r1 = 0x0001                              //write PageAddr high
[P_ECC_Ctrl] = r1

r1 = [P_DMA_Ctrl0]                        // reset ECC
r1 |= 0x0001
[P_DMA_Ctrl0] = r1

?_Write_Not_Finish:                      //enable channel
r1 = [R_DMAFlag]
cmp r1, 1
jne    ?_Write_Not_Finish

r1 = 0x0010
[P_NF_CMD] = r1
call F_CheckNANDBusy
call F_ReadNANDStatus_Byte
.....

IRQ3:
push r1 to [sp]
r1 = [P_INT_Status1]
test r1, C_INT_DMA
jz End_IRQ3

r1=[P_DMA_INT]
[P_DMA_INT]=r1
test  r1,DMA_CH0_INT
jz    End_IRQ3

r1 = 1
[R_DMAFlag] = r1

End_IRQ3:

pop r1 from [sp]
reti

```

18 I2C Controller

18.1 Introduction

The multi-master I2C-bus controller provides a mechanism to communicate between bus masters and peripheral devices by using two signals, a serial data line (SDA) and a serial clock line (SCL). To avoid all possibilities of confusion, data loss and blockage of information, the master and slave devices must have a well-defined protocol.

In multi-master I2C-bus mode, multiple microprocessors can receive or transmit serial data to or from slave devices.

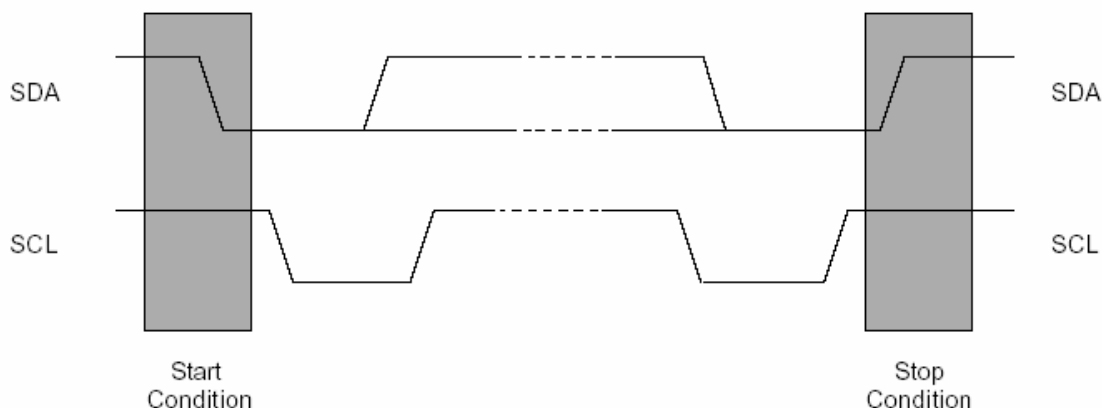
The master that initiates a data transfer over the I2C-bus is responsible for terminating the transfer. It is possible to combine several masters with several slaves onto an I2C-bus to form a multi-master system. If more than one master simultaneously tries to control the bus, an arbitration procedure decides which master gets priority. The maximum number of devices connected to the bus is dictated by the maximum allowable capacitance on the lines, 400 pF.

- Master transmitting and receiving mode
- Slave transmitting and receiving mode
- Detection of bus arbitration failure
- Interrupt generation
- Programmable ACK generation
- Programmable clock speed in master mode
- Input de-bounce circuit

18.2 I2C Bus Protocol

18.2.1 Start / Stop Generation

A “Start” condition means transferring a one-byte serial data over the SDA line, and a “Stop” condition means terminating the data transfer. A “Start” condition is a high-to-low transition of the SDA line while SCL is high. A “Stop” condition is a low-to-high transition of the SDA line while SCL is high. Start and Stop conditions are always generated by the master. The I2C-bus is busy when a Start condition is generated. A few clocks after a Stop condition, the I2C-bus will be free, again.



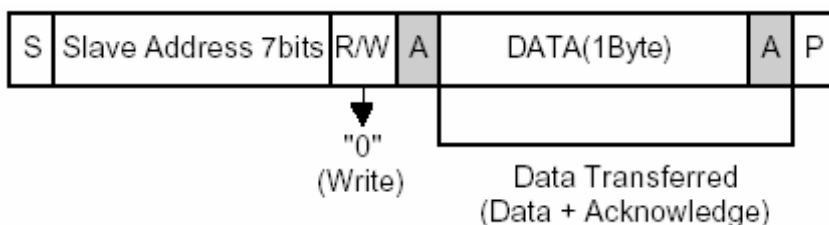
When a master initiates a Start condition, it should send a slave address to notify the slave device. The one byte of address field consists of a 7-bit address and a 1-bit transfer direction indicator (that is, to write or to read). If 8th bit is 0, it indicates a write operation (transmitting operation); if 8th bit is 1, it indicates a request for reading data (receiving operation).

The master will finish a transfer operation by transmitting a Stop condition. If the master wants to continue the data transmission to the bus, it should generate another Start condition as well as a slave address. In this way, the read/write operation can be performed in various formats.

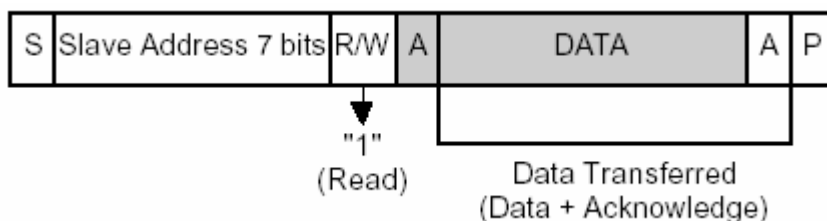
18.2.2 Data Transfer Format

Every byte placed on the SDA line should be eight bits in length. The number of bytes, which can be transmitted per transfer, is unlimited. The first byte following a Start condition should have the address field. The address field can be transmitted by the master when the I2C-bus is operating in master mode. Each byte should be followed by an acknowledgement (ACK) bit. The MSB bit of the serial data and addresses are always sent first.

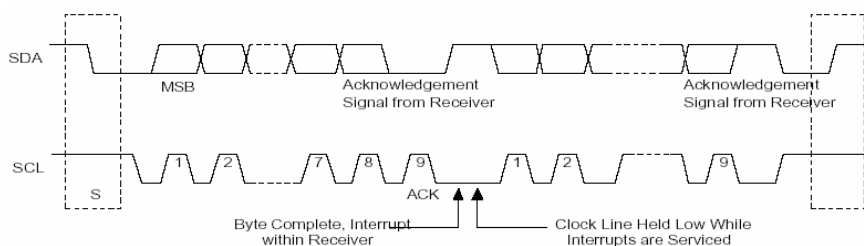
Write mode with 7-bits address



Read mode with 7-bits address



Data transfer on the I2C bus

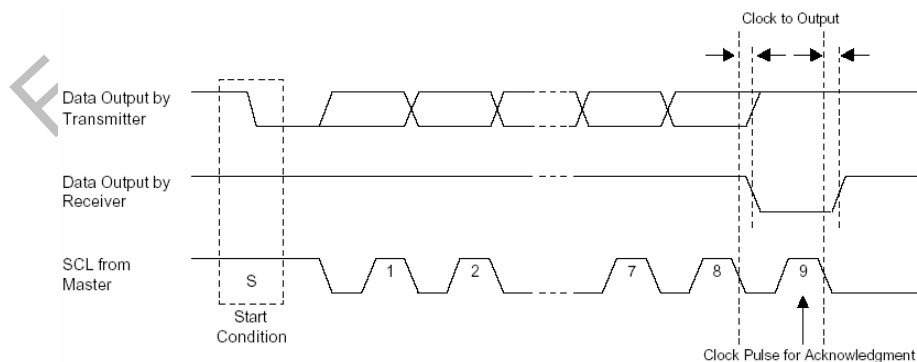


18.2.3 Acknowledgement Signal Transmission

To finish a one-byte transfer operation successfully, the receiver should send an ACK bit to the transmitter. The ACK pulse should occur at the 9th clock of the SCL line. Eight clocks are required for the one-byte data transfer. The master should generate the clock pulse required to transmit the ACK bit.

The transmitter should release the SDA line by making the SDA line High when the ACK clock pulse is received. The receiver should also drive the SDA line Low during the ACK clock pulse so that the SDA is Low during the High period of the 9th SCL pulse.

The ACK bit transmit function can be enabled or disabled by software (P_I2C_Ctrl). However, the ACK pulse on the 9th clock of SCL is required to complete a one-byte data transfer operation.



18.2.4 Read / Write Operation

In the transmitter mode, after the data is transferred, the I2C-bus interface will wait and the SCL line will be low until pending interrupt is cleared. After the interrupt is cleared, the SCL line will be released. After the CPU receives the interrupt request, it should write a new data into P_I2C_Data before clearing the pending interrupt.

In the receiving mode, after a data is received, the I2C-bus interface will wait and the SCL line will be low until pending interrupt is cleared. After the pending interrupt is cleared, the SCL line will be released. After the CPU receives the interrupt request, it should read the data from P_I2C_Data before clear the pending interrupt.

18.2.5 Bus Arbitration Produres

Arbitration takes place on the SDA line to prevent the contention on the bus between two masters. If a master with a SDA high level detects another master with a SDA active Low level, it will not initiate a data transfer because the current level on the bus does not correspond to its own. The arbitration procedure will be extended until the SDA line turns high.

However when the masters simultaneously lower the SDA line, each master should evaluate whether or not the mastership is allocated to itself. For the purpose of evaluation, each master should detect the address bits. While each master generates the slaver address, it should also detect the address bit on the SDA line because the lowering of SDA line is stronger than maintaining high on the line. For example, one master generates a low as first address bit, while the other master is maintaining high. In this case, both masters will detect low on the bus because "Low" is stronger than "High" even if first master is trying to maintain high on the line. When this happens, low-generating (as the first bit of address) master will get the mastership and high-generating (as the first bit of address) master should withdraw the mastership. If both masters generate Low as the first bit of address, there should be arbitration for second address bit, and so on. This arbitration will continue to the end of last address bit.

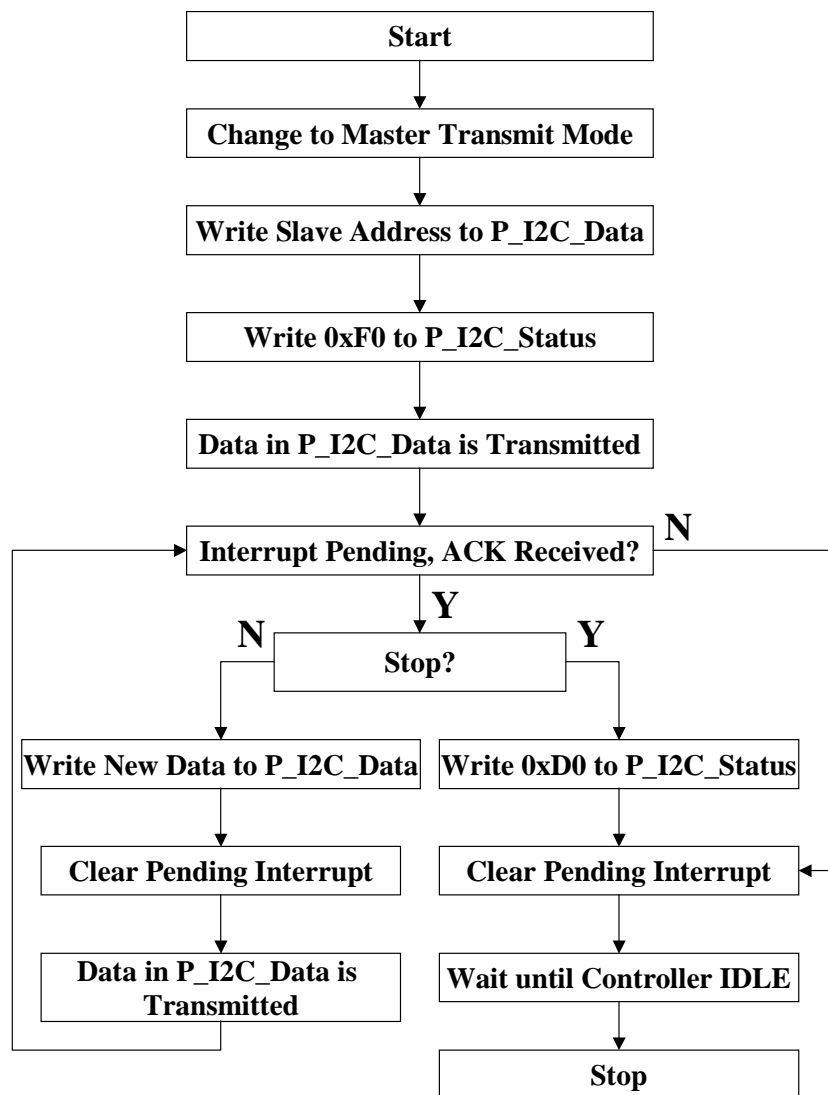
18.2.6 Bus Arbitration Produres

If a slave receiver cannot acknowledge the confirmation of the slave address, it should hold the level of the SDA line High. In this case, the master should generate a Stop condition and to abort the transfer.

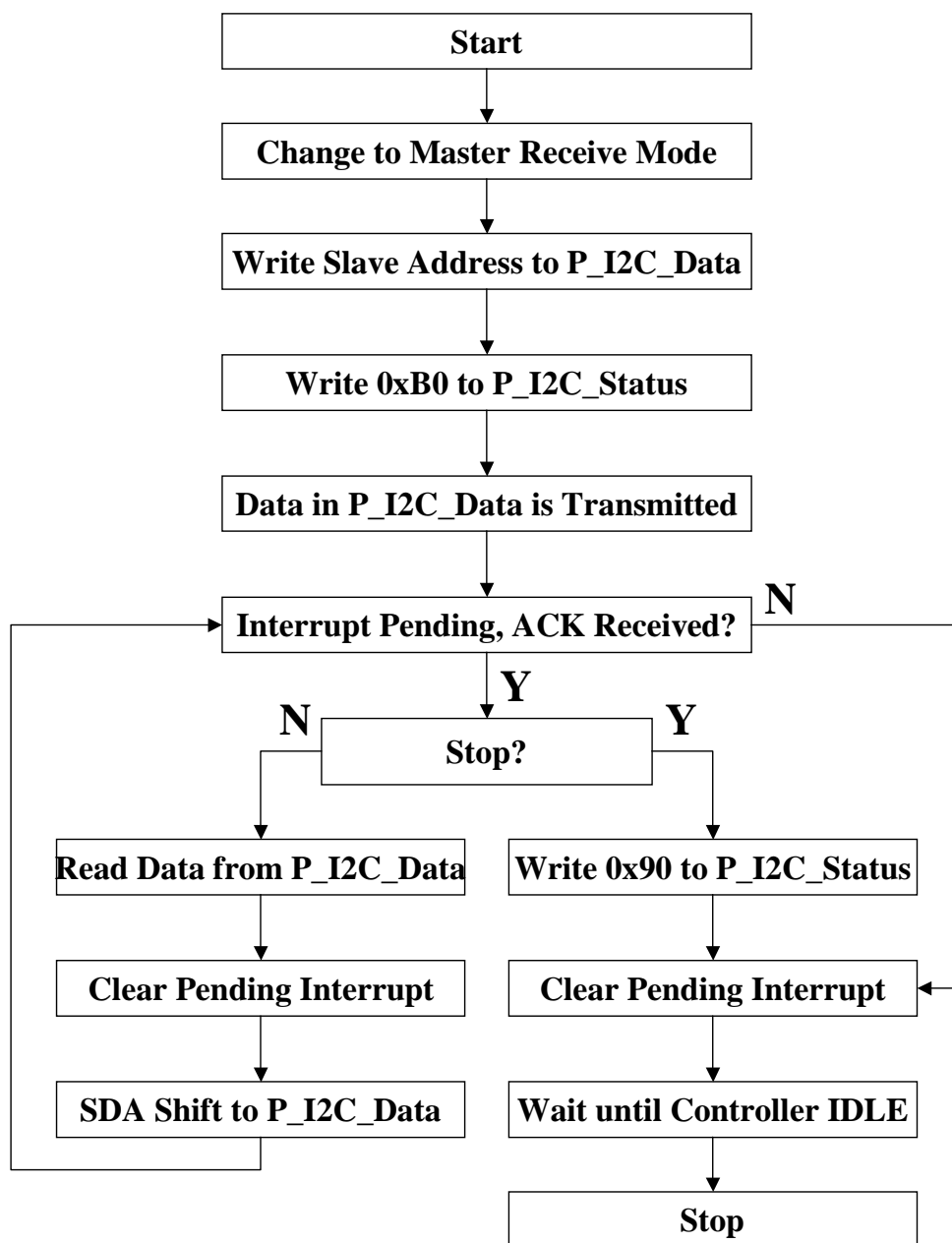
If a master receiver is involved in the aborted transfer, it should signal at the end of the slave transmitting operation by canceling the generation of an ACK after the last data byte is received from the slave. The slave transmitter should then release the SDA to allow a master to generate a Stop condition.

18.3 Firmware Flow Chart

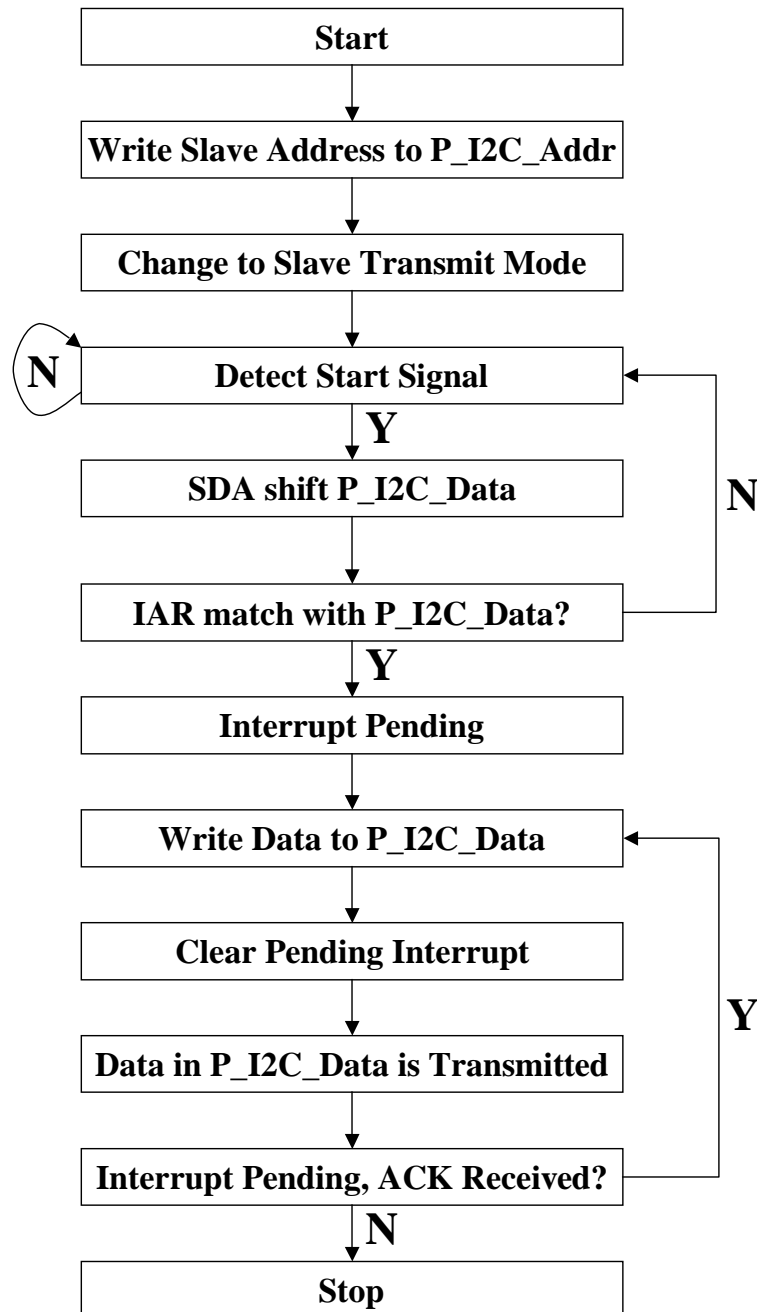
18.3.1 Master Transmit Mode



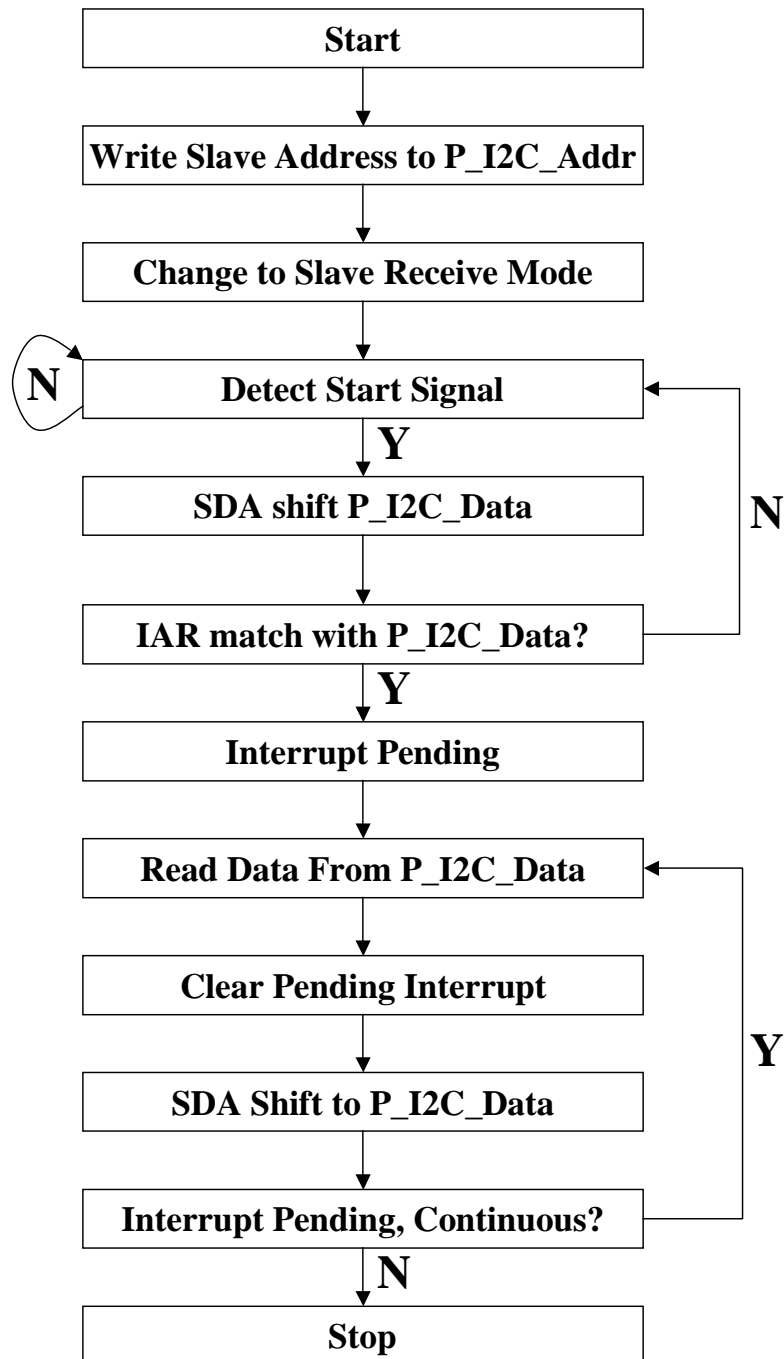
18.3.2 Master Receive Mode



18.3.3 Slave Transmit Mode



18.3.4 Slave Receive Mode



18.4 I2C Bus Control Pin Configuration

Name	I/O	Description
SCK	I/O	I2C Bus SCL input / output (shared with GPIO PortC12)
SDA	I/O	I2C Bus SDA input / output (shared with GPIO PortC13)

18.5 I2C Bus Control Register

I2C Register Summary Table

Name	Address	Description
P_I2C_Ctrl	0x7B60	I2C Control Register
P_I2C_Status	0x7B61	I2C Status Register
P_I2C_Addr	0x7B62	I2C Address Register
P_I2C_Data	0x7B63	I2C Data Register
P_I2C_DeCLK	0x7B64	I2C De-Bounce Clock Register
P_I2C_En	0x7B65	I2C Interface Enable Register

P_I2C_Ctrl				0x7B60				I2C Control Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	-	ACKEN	CLKSEL	INTEN	INTPEND/C	TXCLK				
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
7	ACKEN	R/W	I2C Bus Acknowledge Enable Bit	0= Disable ACK generation 1= Enable ACK generation
6	CLKSEL	R/W	Source Clock of I2C Bus Transmit Clock Prescaler Selection Bit	0= I2CCLK=SysCLK/16 1= I2CCLK=SysCLK/512
5	INTEN	R/W	I2C Bus TX / RX Interrupt Enable If this bit is set to "1", and I2C interrupt is generated, hardware will issue an IRQ5 or FIQ to CPU. If this bit is cleared to "0", the interrupt will be masked off. To select between IRQ5 and FIQ, please refer to Chapter Interrupt .	0= Disable 1= Enable
4	INTPend/C	R/W	I2C Bus TX / RX Interrupt Pending Flag A I2C bus interrupt occurs 1. When a 1-byte transmitting or receiving operation is terminated. 2. When a general call or slave address match occurs. 3. If bus arbitration fails.	Read 0= No interrupt pending Read 1= Interrupt is pending Write 0= No effect Write 1= Clear the flag

Bit	Function	Type	Description	Condition
[3:0]	TXCLK	R/W	<p>I2C Bus Transmit Clock Pre-scaler</p> <p>Transmitting clock frequency is determined by these 4 bits pre-scaler value, according to the following formula:</p> $\text{Tx clock} = \text{I2CCLK} / (\text{P_I2C_Ctrl}[3:0] + 1)$ <p>Where,</p> <ol style="list-style-type: none"> 1. I2CCLK is determined by P_I2C_Ctrl [6] 2. When P_I2C_Ctrl [6] =0, "P_I2C_Ctrl [3:0] =0x00 or 0x01" is not available 	

P_I2C_Status
0x7B61
I2C Status Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Mod	BY	DataEN	ArbS	SS	AddrS	LS	
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:6]	Mod	R/W	<p>I2C Bus Master / Slave, TX / RX Mode Selection</p> <p>These two bits are used to select the master or slave, transmit or receive mode of the I2C bus.</p> <p>Note:</p> <p>Under following 2 kinds of situations, the I2C bus will change to slave receive mode automatically.</p> <ol style="list-style-type: none"> 1. In slave mode, receive slave address is 0x00. 2. In master mode, detects bus arbitration fail. 	<p>00= Slave Receive Mode</p> <p>01= Slave Transmit Mode</p> <p>10= Master Receive Mode</p> <p>11= Master Transmit Mode</p>
5	BY	R/W	<p>I2C Bus Busy Signal Status</p> <p>This bit is used to indicate if the I2C bus is busy or not.</p>	<p>Read 0= Not Busy</p> <p>Read 1= Busy</p> <p>Write 0= I2C Bus interface STOP signal generation</p> <p>Write 1=I2C Bus interface START signal generation</p>
4	DataEn	R/W	<p>I2C Bus Data Output Enable</p> <p>If this bit is set to "1", I2C data output is enabled. Or, the data output is disabled.</p>	<p>0= Disable RX/TX</p> <p>1= Enable RX/TX</p>
3	ArbS	R	<p>I2C Bus Arbitration Procedure Status Flag</p> <p>This bit is used to indicate if the arbitration procedure is okay or not.</p>	<p>0= Bus arbitration status okay</p> <p>1= Bus arbitration failed</p>
2	SS	R	I2C Bus Address-as-Slave Status Flag	0= START / STOP condition

Bit	Function	Type	Description	Condition
				was generated 1= Received slave address matches the address value in the P_I2C_Addr
1	AddS	R	I2C Bus Address Zero Status Flag	0= START / STOP condition was generated 1= Received slave address is 0x0
0	LS	R	I2C Bus Last-Received Bit Status Flag	0= Last-received bit is "0" (ACK was received) 1= Last-received bit is "1" (ACK was not received)

P_I2C_Addr
0x7B62
I2C Address Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:1]	Addr	R/W	I2C Bus Address 7-bit slave address, latched from the I2C bus: When data output enable bit="0" in the P_I2C_Status [4], it is able to write P_I2C_Addr. It is allowable to read this register at any time, regardless of the current serial output enable bit, P_I2C_Status[4].	[7:1]=Slave Address
0			Reserved	

P_I2C_Data
0x7B63
I2C Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data	R/W	I2C Data Register 8-bit data shift register for I2C bus TX / RX operation:	

Bit	Function	Type	Description	Condition
			When serial output enable="1" in the P_I2C_Status, it is able to write this register. It is allowable to read this register value at any time, regardless of the current serial output enable bit, P_I2C_Status[4].	

P_I2C_DeCLK 0x7B64 I2C De-bounce Clock Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DEBCLK							
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	DEBCLK	R/W	De-bounce Clock Since rising time of I2C bus is very slow, this register is used to de-bounce the input signal on I2C bus. I2C input signal will be latched every DEBCLK cycles of system clock.	0~255= 0~255 cycles of system clock

P_I2C_En 0x7B65 I2C Enable Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2CEN
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:1]			Reserved	
0	I2CEN	R/W	I2C Bus Enable If this bit is set to "1", I2C interface is enabled. And IOC [13:12] cannot be used as GPIOs.	0= Disable 1= Enable

18.6 Example Program

```

R1 = 0x01                                // Enable I2C interface
[P_I2C_En] = r1
r1=0x09a                                //clock is 75kHz, enable ACK
[P_I2C_Ctrl]=r1

r1 = 0x0d0                                //master tx, enable rx/tx
[P_I2C_Status]=r1

r1=0x010
[P_I2C_DeCLK]=r1

r1=Device_addr                            //send device addr
[P_I2C_Data]=r1
r1=0x0f0                                // Write 0x00f0 to status register
[P_I2C_Status]=r1

waitloop?:
r1=[P_I2C_Ctrl]
test r1,0x010
jz waitloop?
r1=[P_I2C_Status]
r1&=0x0f
test r1,0x01
jz next?
goto F_Error                                // Did not receive ACK

next?:
test r1,0x08
jz next1?
goto F_Error                                // Bus arbitration failed

r1 = Send_Data                            // send data
[P_I2C_Data] = r1

r1=[P_I2C_Ctrl]                            //clear int flag and send data
[P_I2C_Ctrl]=r1

waitloop2?:
r1=[P_I2C_Ctrl]
test r1,0x010
jz waitloop2?
r1=[P_I2C_Status]
r1&=0x0f

```

```
test r1,0x01
jz next4?
Goto F_Error

next4?:

test r1,0x08
jz Master_Transmit_Complete?
r1=0x0d0
[P_I2C_Status]=r1
goto Error

Master_Transmit_Complete?:

jmp $
```

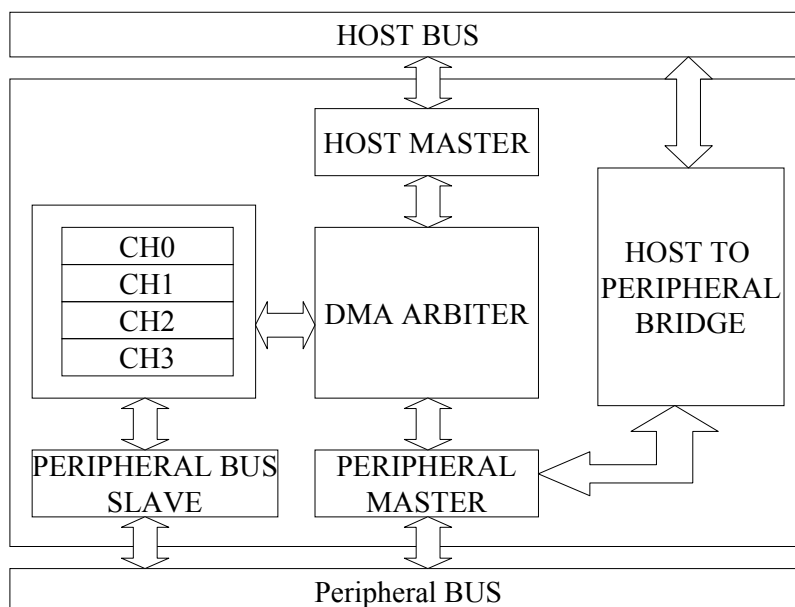
19 DMA and Bridge Controller

19.1 Introduction

The DMA controller built in GPL162002A/162003A is a 4-channel DMA controller combined with a host-to-peripheral bridge. Each DMA channel is capable of doing DMA transfer from any memory address to another memory address. To minimize the host bus usage, the DMA channel is also able to do the DMA transfer from IO to memory or from memory to IO or even from IO to IO. During the DMA transfer, the DMA controller can do the byte-to-word or word-to-byte conversion for the simplification of software coding.

- 4 independent DMA channels
- Both host bus master and peripheral bus master are integrated
- Memory-to-memory, memory-to-IO, IO-to-memory and IO-to-IO modes are available
- Both external request and software request modes are available
- 26-bit addressing
- 8-bit and 16-bit peripheral supported
- Support double buffer mode
- Support pattern match transparent function
- Support sprite auto move function
- Support DMA time-out interrupt
- Integrate Host-to-peripheral bridge

19.2 Block Diagram



19.3 Operation Mode

There are two operation modes when accessing DMA: Software mode and External mode.

When the DMA channel is operating in software mode, the DMA will start immediately as soon as the CE in P_DMA_Ctrl [0] is 1 and P_DMA_TCount > 0. It will continue moving the data until the P_DMA_TCount counts to 0.

When the DMA channel operates in external mode, the DMA channel will wait for the external request signal to initiate a read/write sequence. There are two modes in detecting the external request signal: Single mode and Demand mode, respectively.

19.3.1 Single Mode

The single mode means only the rising edge of the external request will be detected. Each rising edge of the request signal will result in a single DMA transfer.

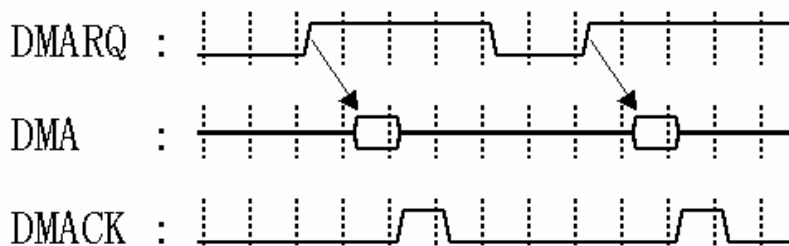


Fig1. External Single mode

19.3.2 Demand Mode

The demand mode means the DMA request become level sensitive. If the DMA request is detected high at the beginning, the DMA controller will start a DMA transfer. At the end of the DMA transfer, DMA controller will detect the level of the DMA request signal; if the DMA request signal is still high, another burst read/write will start. Other DMA channel could not get the bus grant if the DMA request is kept low during a demand transfer mode, but if the P_DMA_TCount reach 0, the bus grant could be released even the DMA request is still low. The peripheral built in GPL162002A/162003A, such as SD Card, UART ...etc., are all using this mode.

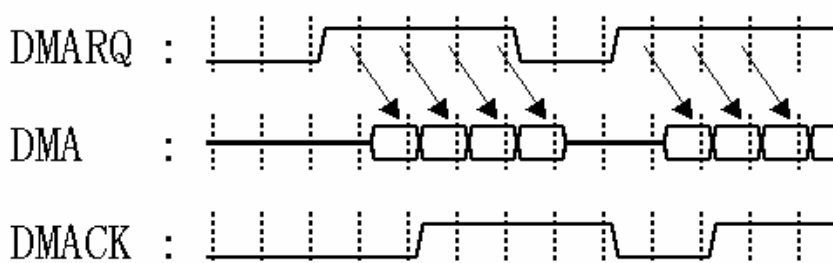


Fig2. External Demand Mode

19.4 Double Buffer Mode

In order to prevent the gap between a DMA transfer and another DMA transfer, the DMA controller provides the double buffer mode. With the double buffer mode, programmers can write the new address/count of next DMA transfer before the end of current DMA transfer. The DMA channel will begin the next DMA transfer immediately after current DMA transfer is completed. This is very useful in playing music and recording voice. But if programmers write new address/count when there is already a queue in the DMA channel, the old one will be overwritten, so cares must be taken to ensure the DBF bit in P_DMA_Ctrl is 0 when writing the new address/count.

The sequence of writing a new address/count is P_DMA_SRC_AddrL => P_DMA_SRC_AddrH => P_DMA_TAR_AddrL => P_DMA_TAR_AddrH => P_DMA_TCountL => P_DMA_TCountH. Programmers must write P_DMA_TCountL/P_DMA_TCountH at the end of the updating sequence. The P_DMA_Ctrl does not need to be updated since only the DMA transfer with the same configuration can use the double buffer mode.

19.5 Byte Mode Operation

When SRCBYTE in P_DMA_Ctrl is set to 1, the DMA channel will treat the data read from a source in byte. If SF in P_DMA_Ctrl is 0, the P_DMA_SRC_Addr will increase/decrease after every “two” readings from a source. When TARBYTE in P_DMA_Ctrl is set to 1, the DMA channel will write data to a peripheral in byte. If DF in P_DMA_Ctrl is 0, the P_DMA_SRC_Addr will increase/decrease after every “two” writings to destination. If WRITEREQ in P_DMA_Ctrl is 0, this means that the request from a peripheral needs to be read, so the P_DMA_TCount will decrease at every reading to a peripheral. If WRITEREQ in P_DMA_Ctrl is 1, this means that the request from a peripheral needs to be written, so the P_DMA_TCount will decrease at every writing to peripheral.

For example, the DMA needs to read from SPI and write to a memory device for 512 bytes (IO to memory). With SRCBYTE=1, TARBYTE=0, WRITEREQ=0, and P_DMA_TCountL=512, The DMA will read one byte from SPI while every request comes, and P_DMA_TCount will be decreased by one. While every two readings complete, the DMA will cascade these two bytes in one word and write to a memory device. For other conditions, please refer to the following two tables.

Table 1 Behavior of DMA controller when WRITEREQ = 0

Condition	Read Start	P_DMA_SRC_Addr Change.	Write Start	P_DMA_TAR_Addr Change	P_DMA_TCount Decrease
SRCBYTE = 0 TARBYTE = 0	Request Come.	Read Complete.	Read Complete.	Write Complete.	Read Complete.
SRCBYTE = 1 TARBYTE = 0	Each request results in one read.	2 Reads Complete.	2 Reads Complete or P_DMA_TCount is zero.	Write Complete.	1 Read Complete.
SRCBYTE = 0 TARBYTE = 1	Request Come.	Read Complete.	Read Complete. Two writes are continues.	2 Writes Complete.	Read Complete.
SRCBYTE = 1 TARBYTE = 1	Each request result in one read.	2 Reads Complete.	2 Reads Complete or P_DMA_TCount is zero. Two writes are continues.	2 Writes Complete.	1 Read Complete.

Table 2 Behavior of DMA controller when WRITEREQ = 1

Condition	Read Start	P_DMA_SRC_Addr Change.	Write Start	P_DMA_TAR_Addr Change	P_DMA_TCount Decrease
SRCBYTE = 0 TARBYTE = 0	Request Come.	Read Complete.	Read Complete.	Write Complete.	Write Complete.
SRCBYTE = 1 TARBYTE = 0	Each request results in two read.	2 Reads Complete.	2 Reads Complete or P_DMA_TCount is zero.	Write Complete.	Write Complete.
SRCBYTE = 0 TARBYTE = 1	Request Come.	Read Complete.	Read Complete. Each request results in one write. Data will be hold until another request come.	2 Writes Complete.	1 Write Complete.
SRCBYTE = 1 TARBYTE = 1	Each request results in two read.	2 Reads Complete.	2 Reads Complete or P_DMA_TCount is zero. Each request results in one write. Data will be hold until another request come.	2 Writes Complete.	1 Write Complete.

19.6 Control Register

DMA Control Register Summary Table

Name	Address	Description
P_DMA_Ctrl0	0x7B80	DMA Channel Control Register 0
P_DMA_SRC_AddrL0	0x7B81	DMA Source Low Address [15:0] Register 0
P_DMA_TAR_AddrL0	0x7B82	DMA Target Low Address [15:0] Register 0
P_DMA_TCountL0	0x7B83	DMA Terminal Counter Low [15:0] Register 0
P_DMA_SRC_AddrH0	0x7B84	DMA Source High Address [25:16] Register 0
P_DMA_TAR_AddrH0	0x7B85	DMA Target High Address [25:16] Register 0
P_DMA_TCountH0	0x7B86	DMA Terminal Counter High [25:16] Register 0
P_DMA_MISC0	0x7B87	DMA miscellaneous Control Register 0
P_DMA_Ctrl1	0x7B88	DMA Channel Control Register 1
P_DMA_SRC_AddrL1	0x7B89	DMA Source Low Address [15:0] Register 1
P_DMA_TAR_AddrL1	0x7B8A	DMA Target Low Address [15:0] Register 1
P_DMA_TCountL1	0x7B8B	DMA Terminal Counter Low [15:0] Register 1
P_DMA_SRC_AddrH1	0x7B8C	DMA Source High Address [25:16] Register 1
P_DMA_TAR_AddrH1	0x7B8D	DMA Target High Address [25:16] Register 1
P_DMA_TCountH1	0x7B8E	DMA Terminal Counter High [25:16] Register 1
P_DMA_MISC1	0x7B8F	DMA miscellaneous Control Register 1
P_DMA_Ctrl2	0x7B90	DMA Channel Control Register 2
P_DMA_SRC_AddrL2	0x7B91	DMA Source Low Address [15:0] Register 2
P_DMA_TAR_AddrL2	0x7B92	DMA Target Low Address [15:0] Register 2

Name	Address	Description
P_DMA_TCountL2	0x7B93	DMA Terminal Counter Low [15:0] Register 2
P_DMA_SRC_AddrH2	0x7B94	DMA Source High Address [25:16] Register 2
P_DMA_TAR_AddrH2	0x7B95	DMA Target High Address [25:16] Register 2
P_DMA_TCountH2	0x7B96	DMA Terminal Counter High [25:16] Register 2
P_DMA_MISC2	0x7B97	DMA miscellaneous Control Register 2
P_DMA_Ctrl3	0x7B98	DMA Channel Control Register 3
P_DMA_SRC_AddrL3	0x7B99	DMA Source Low Address [15:0] Register 3
P_DMA_TAR_AddrL3	0x7B9A	DMA Target Low Address [15:0] Register 3
P_DMA_TCountL3	0x7B9B	DMA Terminal Counter Low [15:0] Register 3
P_DMA_SRC_AddrH3	0x7B9C	DMA Source High Address [25:16] Register 3
P_DMA_TAR_AddrH3	0x7B9D	DMA Target High Address [25:16] Register 3
P_DMA_TCountH3	0x7B9E	DMA Terminal Counter High [25:16] Register 3
P_DMA_MISC3	0x7B9F	DMA miscellaneous Control Register 3
P_DMA_SPRISIZE0	0x7BB0	DMA Sprite Size [9:0] Register 0
P_DMA_SPRISIZE1	0x7BB1	DMA Sprite Size [9:0] Register 1
P_DMA_SPRISIZE2	0x7BB2	DMA Sprite Size [9:0] Register 2
P_DMA_SPRISIZE3	0x7BB3	DMA Sprite Size [9:0] Register 3
P_DMA_TRANSPAT0	0x7BB8	DMA Transparent Pattern Register 0
P_DMA_TRANSPAT1	0x7BB9	DMA Transparent Pattern Register 1
P_DMA_TRANSPAT2	0x7BBA	DMA Transparent Pattern Register 2
P_DMA_TRANSPAT3	0x7BBB	DMA Transparent Pattern Register 3
P_DMA_LINELENGTH	0x7BBD	DMA Line Length Control Register
P_DMA_SS	0x7BBE	DMA Source Select Register
P_DMA_INT	0x7BBF	DMA Interrupt Status Register

P_DMA_Ctrl0		0x7B80				DMA Channel Control Register 0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE	
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_Ctrl1		0x7B88						DMA Channel Control Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE	
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_Ctrl2		0x7B90						DMA Channel Control Register 2								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_Ctrl3		0x7B98						DMA Channel Control Register 3									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15	WRITEREQ	R/W	Peripheral Write Request (Byte Mode) Indicate that the request from a peripheral needs to be written or read. This bit is only valid when SRCBYTE or TARBYTE is 1 and MODE is 1 (External Mode).	0x0= Request coming from a peripheral whose data need to be read. 0x1= Request coming from a peripheral whose data need to be written.
14	TM	R/W	Transfer Mode Field This bit is used to indicate single transfer mode or demand transfer mode. Note that this bit is only valid when MODE is set to 1 (external mode).	0x0= Single transfer mode. 0x1= Demand transfer mode.
13	TARBYTE	R/W	Target Byte Selection Indicate if the target is in byte mode or not.	0x0= Target is 16-bits mode. 0x1= Target is 8-bits mode.
12	SRCBYTE	R/W	Source Byte Selection Indicate if the source is in byte mode or not.	0x0= Source is 16-bits mode. 0x1= Source is 8-bits mode.
11:10	TD	R/W	Transfer direction field. These two bits are used to select DMA transfer direction.	00= Memory to Memory 01= Memory to IO 10= IO to Memory 11= IO to IO
9	RS	W	Software reset If this bit is set to "1", the values of control register in this channel will be reset to default.	Write 1= Reset control register.
8	CIE	R/W	Channel Interrupt Enable If this bit is set to "1", and if P_DMA_TCount reaches 0, hardware will issue an IRQ3 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ3 or FIQ, please refer to Chapter Interrupt	0= Disable DMA interrupt 1= Enable DMA interrupt
7	SF	R/W	Source Fixing Address If this bit is set to "1", the source address will be fixed when data is read from the source address.	0= Increase/decrease source address. 1= Fix source address
6	DF	R/W	Destination Fixing Address If this bit is set to "1", the destination	0= Increase/decrease destination address.

Bit	Function	Type	Description	Condition
			address will be fixed when data is written to the destination address.	1= Fix destination address.
5	SD	R/W	Source Address Direction If this bit is set to "1", the source address will be decreased when data is read from the source address. Note that this bit will be valid only when SF is 0.	0= Increase address. 1= Decrease address.
4	DD	R/W	Destination Address Direction If this bit is set to "1", the destination address will be decreased when data is written to the destination address. Note that this bit will be valid only when DF is 0.	0= Increase address. 1= Decrease address.
3	DBF	R	DMA Double Buffer Full. When DMA is active and programmers write the P_DMA_TCountL/H again, this bit will be set to 1. The value of P_DMA_SRC_AddrL/H and P_DMA_TAR_AddrL/H can be updated before writing to P_DMA_TCountL/H. When the current DMA action is completed, it will automatically reload the value in these three registers and perform the next DMA transfer.	1= Occur 0= Not occur
3	NORM_I	W	DMA Normal Interrupt Mode This bit is used to set up the interrupt mode.	0x0= DMA issue interrupt only when P_DMA_TCount reach 0 and DBF is 0. 0x1= DMA issue interrupt every time when DMATCR reach 0 and don't care the DBF.
2	MODE	R/W	DMA mode selection. This bit is used to select DMA operation mode. In Software mode, DMA transfer will start automatically until P_DMA_TCount reaches 0. In External mode, DMA controller will not initialize a DMA transfer until acquiring a DMA request from a peripheral.	0= Software mode 1= External mode
1	BS	R	Status of DMA Channel.	0= Idle 1= Busy
0	CE	R/W	Channel Enable	0= Channel is disabled. 1= Channel is enabled.

P_DMA_SRC_AddrL0 0x7B81 DMA Source Low Address Register 0																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SRC_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrL1				0x7B89				DMA Source Low Address Register 1								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SRC_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrL2				0x7B91				DMA Source Low Address Register 2								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SRC_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrL3 0x7B99 DMA Source Low Address Register 3																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	SRC_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	SRC_Addr	R/W	DMA Source Low Address [15:0]	

These P_DMA_SRC_AddrLx registers are low address [15:0] registers of sources. The value in these registers will be increased/decreased when a word is read and when the SF in P_DMA_Ctrl [7] is 0. It should be noted if the TD in P_DMA_Ctrl[11:10] is set as IO to memory or IO-to-IO mode, only the lower 12 bits will be used to issue a peripheral read.

DMA Target Low Address Register 0																
P_DMA_TAR_AddrL0	0x7B82															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TAR_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrL1				0x7B8A				DMA Target Low Address Register 1								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TAR_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrL2				0x7B92				DMA Target Low Address Register 2									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TAR_Addr																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TAR_AddrL3				0x7B9A				DMA Target Low Address Register 3											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	TAR_Addr																		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

Bit	Function	Type	Description	Condition
[15:0]	TAR_Addr	R/W	DMA Target Low Address [15:0]	

These P_DMA_TAR_AddrLx registers are low address [15:0] registers of targets. The value in these registers will be increased/decreased when a word is written and when the DF in P_DMA_Ctrl [6] is 0. It should be noted if the TD in P_DMA_Ctrl[11:10] is set to memory-to-IO or IO-to-IO mode, only the lower 12 bits will be used to issue a peripheral write.

P_DMA_TCountL0		0x7B83								DMA Terminal Count Low Register 0							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TCountL															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL1		0x7B8B								DMA Terminal Count Low Register 1							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TCountL															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL2		0x7B93								DMA Terminal Count Low Register 2							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TCountL															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL3				0x7B9B				DMA Terminal Count Low Register 3								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TCountL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	TCountL	R/W	DMA Terminal Count Low Address [15:0]	

The P_DMA_TCountLx registers are the terminal count low registers for each DMA channel and contain 16-bits value of remaining number of DMA transfers. The DMA transfer will start only when this register is not zero. It should be noticed that the value in this register means the remaining DMA transfer counts not the remaining words to be transferred. The number in these registers will be decreased by one when a DMA transfer is completed.

When the WRITEREQ in P_DMA_Ctrl [15] is set to 1, only the writing operation will decrease the counter. Nevertheless, when the WRITEREQ in P_DMA_Ctrl [15] is set to 0, only the reading operation will decrease the counter. This is only valid when SRCBYTE in P_DMA_Ctrl [12] or TARBYTE in P_DMA_Ctrl [13] is set to 1.

P_DMA_SRC_AddrH0				0x7B84			DMA Source High Address Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH1				0x7B8C			DMA Source High Address Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH2							DMA Source High Address Register 2										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	SRC_AddrH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_SRC_AddrH3			0x7B9C				DMA Source High Address Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reservede	
[9:0]	SRC_AddrH	R/W	DMA Source High Address [25:16]	

The P_DMA_SRC_AddrHx registers are the source high address [25:16] registers. The value in these registers will be increased/decreased when a word is read and the SF in P_DMA_Ctrl [7] is 0. It should be noted if the TD in P_DMA_Ctrl [11:10] is set as IO to memory or IO-to-IO mode, this register is useless.

P_DMA_TAR_AddrH0		0x7B85						DMA Target High Address Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TAR_AddrH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TAR_AddrH1		0x7B8D						DMA Target High Address Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TAR_AddrH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TAR_AddrH2				0x7B95				DMA Target High Address Register 2									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TAR_AddrH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TAR_AddrH3				0x7B9D				DMA Target High Address Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TAR_AddrH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	TAR_AddrH	R/W	DMA Target High Address [25:16]	

The P_DMA_TAR_AddrHx registers are the destination high address [25:16] registers. The value in these registers will be increased/decreased when a word is written and the DF in P_DMA_Ctrl [6] is 0. It should be noted if the TD in P_DMA_Ctrl [11:10] is set to memory-to-IO or IO-to-IO mode, this register is useless.

P_DMA_TCountH0		0x7B86						DMA Terminal Count High Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TCountH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TCountH1			0x7B8E					DMA Terminal Count High Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TCountH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TCountH2		0x7B96						DMA Terminal Count High Register 2									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	TCountH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountH3		0x7B9E						DMA Terminal Count High Register 3									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	TCountH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	TCountH	R/W	DMA Terminal Count High [25:16]	

P_DMA_MISC0		0x7B87				DMA miscellaneous Control Register 0											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DMARQ	ERRW	-	TRANS_EN	DMATO								-		STATE		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_MISC1		0x7B8F					DMA miscellaneous Control Register 1										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_MISC2		0x7B97				DMA miscellaneous Control Register 2											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_MISC3				0x7B9F				DMA miscellaneous Control Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
15	DMARQ	R	DMA Terminal Count High [25:16]	
14	ERRW	R	CPU update the DMACONX when a DMA is bust	
13			Reserved	
12	TRANS_EN	R/W	DMA Transparent Enable If this bit is set to 1, then when DMA read a data matched the TRANSPART, this data will not be written to target address.	0: Disable 1: Enable

Bit	Function	Type	Description	Condition
[11:4]	DMATO	R/W	DMA Time Out Counter These bits are to set the time out count of the DMA. When P_DMA_Ctrl.b0 is set to 1, the timer starts to count. And If the DMA controller does not complete data transmission at duration of designated time out count, then hardware will issue a DMA interrupt to CUP.	00: Time-out function disable 01: DMA will time-out in 1/256 sec 02: DMA will time-out in 2/256 sec 03: DMA will time-out in 3/256 sec FF: DMA will time-out in 255/256 sec
3			Reserved	
[2:0]	STATE	R	DMA Controller's State register	

P_DMA_SPRISIZE0			0x7BB0			DMA Sprite Size [9:0] Register 0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SPRISIZE									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE1			0x7BB1			DMA Sprite Size [9:0] Register 1										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SPRISIZE									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE2			0x7BB2			DMA Sprite Size [9:0] Register 2										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SPRISIZE									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE3			0x7BB3			DMA Sprite Size [9:0] Register 3										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SPRISIZE									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	SPRISIZE	R/W	Sprite Size of DMA Channel	00: Close sprite mdoe.

This register is used to determine the sprite size in X-axis. The sprite size must be smaller than LINELENGTH (0x7BBD.b [9:0]). When DMA has transferred data amount equal to SPRISIZE, the target address will increase automatically by (LINELENGTH – SPRISIZE). This function is useful by moving

data into LCD frame buffer. When SPRUSIZE is set to 0, the sprite mode will be closed. Only 16-bit transfer mode is supported in sprite mode and the DF/DD must equal to 0 in this mode.

P_DMA_TRANSPAT0		0x7BB8								DMA Transparent Pattern Register 0							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TRANSPAT																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TRANSPAT1		0x7BB9		DMA Transparent Pattern Register 1													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TRANSPAT																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TRANSPAT2				0x7BBA				DMA Transparent Pattern Register 2											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	TRANSPAT																		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_DMA_TRANSPAT3				0x7BBB				DMA Transparent Pattern Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TRANSPAT																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

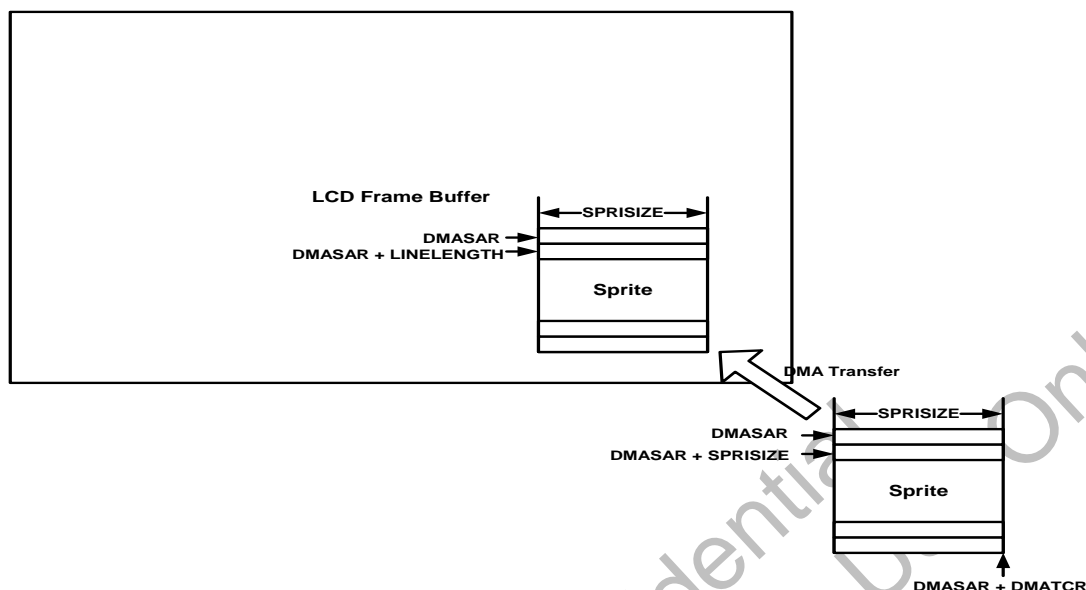
Bit	Function	Type	Description	Condition
[15:0]	TRANSPAT	R/W	Transparent Pattern of DMA Channel	

This register is used to determine the transparent pattern in DMA transfer. When TRANSPEN (0x7B87.b12) set to 1 and DMA read data matched the value stored in this register, the read data will not be written to the destination address.

P_DMA_LINELENGTH		0x7BBD						DMA Line Length Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	LINELENGTH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:10]			Reserved	
[9:0]	LINELENGTH	R/W	Line Length of DMA Sprite	

The following diagram show the memory mapping when sprite mode is turned on.



P_DMA_SS				0x7BBE				DMA Source Select Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DMA_SS3				DMA_SS2				DMA_SS1				DMA_SS0			
Init	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]	DMA_SS3	R/W	DMA Channel 3 Source Select R/W	0x0: USB 0x1: DAC CHA 0x2: UART TX 0x3: UART RX 0x4: SD/MMC 0x5: NAND Flash 0x6: Serial Interface 0x7: DAC CHB 0x8: ADC Auto Sample Full 0x9: SPI TX 0xA: SPI RX Other: Reserved
[11:8]	DMA_SS2	R/W	DMA Channel 2 Source Select R/W	0x0: USB 0x1: DAC CHA 0x2: UART TX 0x3: UART RX 0x4: SD/MMC 0x5: NAND Flash 0x6: Serial Interface 0x7: DAC CHB

Bit	Function	Type	Description	Condition
				0x8: ADC Auto Sample Full 0x9: SPI TX 0xA: SPI RX Other: Reserved
[7:4]	DMA_SS1	R/W	DMA Channel 1 Source Select R/W	0x0: USB 0x1: DAC CHA 0x2: UART TX 0x3: UART RX 0x4: SD/MMC 0x5: NAND Flash 0x6: Serial Interface 0x7: DAC CHB 0x8: ADC Auto Sample Full 0x9: SPI TX 0xA: SPI RX Other: Reserved
[3:0]	DMA_SS0	R/W	DMA Channel 0 Source Select R/W	0x0: USB 0x1: DAC CHA 0x2: UART TX 0x3: UART RX 0x4: SD/MMC 0x5: NAND Flash 0x6: Serial Interface 0x7: DAC CHB 0x8: ADC Auto Sample Full 0x9: SPI TX 0xA: SPI RX Other: Reserved

Each DMA channel has its own DMA request and DMA acknowledge signal. These signals do not have to be connected to specific peripherals. In other words, a DMA channel could receive a DMA request signal determined by setting P_DMA_SS register.

P_DMA_INT					0x7BBF											DMA Interrupt Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
Function	-	-	-	-	CH3BY	CH2BY	CH1BY	CH0BY	CH3TOIF	CH2TOIF	CH1TOIF	CH0TOIF	CH3IF	CH2IF	CH1IF	CH0IF									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									

Bit	Function	Type	Description	Condition
[15:12]			Reserved	

Bit	Function	Type	Description	Condition
11	CH3BY	R	DMA Channel3 Busy Flag	0= Idle 1= Busy
10	CH2BY	R	DMA Channel2 Busy Flag	0= Idle 1= Busy
9	CH1BY	R	DMA Channel1 Busy Flag	0= Idle 1= Busy
8	CH0BY	R	DMA Channel0 Busy Flag	0= Idle 1= Busy
7	CH3TOIF	R	DMA Channel 3 Time Out flag This bit will be clear if programmers write 1 to CH3IF to clear interrupt flag	0= Not time out interrupt 1= Time out interrupt
6	CH2TOIF	R	DMA Channel 2 Time Out flag This bit will be clear if programmers write 1 to CH2IF to clear interrupt flag	0= Not time out interrupt 1= Time out interrupt
5	CH1TOIF	R	DMA Channel 1 Time Out flag This bit will be clear if programmers write 1 to CH1IF to clear interrupt flag	0= Not time out interrupt 1= Time out interrupt
4	CH0TOIF	R	DMA Channel 0 Time Out flag This bit will be clear if programmers write 1 to CH0IF to clear interrupt flag	0= Not time out interrupt 1= Time out interrupt
3	CH3IF	R/W	DMA Channel 3 complete Interrupt Flag	Read 0= Not Occur Read 1= Occur Write 0= No Effect Write 1= Clear the Flag
2	CH2IF	R/W	DMA Channel 2 complete Interrupt Flag	Read 0= Not Occur Read 1= Occur Write 0= No Effect Write 1= Clear the Flag
1	CH1IF	R/W	DMA Channel 1 complete Interrupt Flag	Read 0= Not Occur Read 1= Occur Write 0= No Effect Write 1= Clear the Flag
0	CH0IF	R/W	DMA Channel 0 complete Interrupt Flag	Read 0= Not Occur Read 1= Occur Write 0= No Effect Write 1= Clear the Flag

19.7 Program Examples

Memory to memory:

```
r1=0x0200                // DMA channel Reset
[P_DMA_Ctrl0]=r1

r1=0x00                  // Set source address to 0x30000
[P_DMA_SRC_AddrL0]=r1
r1=0x03
[P_DMA_SRC_AddrH0]=r1

r1=0x5000                // Set target address to 0x5000
[P_DMA_TAR_AddrL0]=r1
r1=0x00
[P_DMA_TAR_AddrH0]=r1

r1=0x1902                // Transfer length 0x1902
[P_DMA_TCountL0]=r1
r1=0x00
[P_DMA_TCountH0]=r1

r1=0x4009                // Single transfer mode, target & source 16 bit,
[P_DMA_Ctrl0]=r1          // memory to memory, disable DMA interrupt,
                          // source address & target address increase
                          // DMA issue interrupt when P_DMA_TCount
                          // reach 0 and don't care the DBF, software
                          // mode
```

Read_Finish?

```
r1=[P_DMA_INT]
test r1, 0x01
jz Read_Finish?

r1=0x01
[P_DMA_INT]=r1

jmp $
```

20 SD and MMC Memory/IO Card Controller

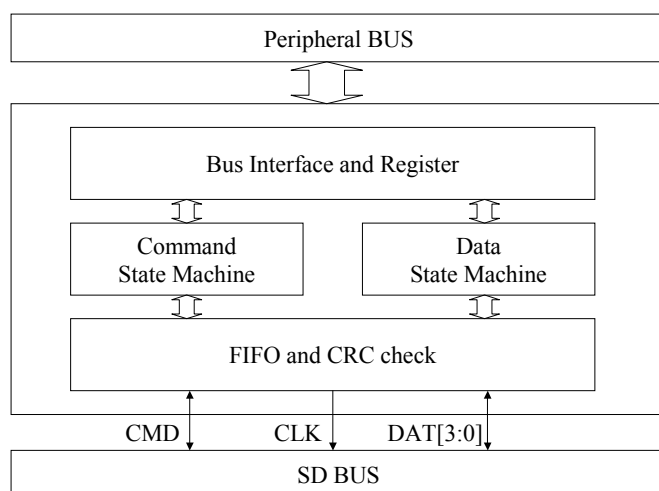
20.1 Introduction

Secure Digital (SD) memory card is a Flash-based memory card that is specifically designed to meet the security, capacity, performance and environment requirements inherent in newly emerging audio and video consumer electronic devices. The SD Memory Card communication is based on an advanced 9-pin interface (Clock, Command, 4xData and 3xPower lines) designed to operate in a low voltage range. SD IO card controller is based on and is compatible with the SD memory card. The intent of the SD IO card controller is to provide high-speed data I/O with low power consumption for mobile electronic devices.

The SD/MMC card controller built in GPL162002A/162003A is designed to have high performance transfer rate by using DMA access, which can achieve the best performance/cost ratio.

- Fully compatible with SD/MMC Memory card specification
- Accept SD commands directly to improve the compatibility
- Programmable clock speed on the SD bus
- SD bus clock control while the buffer is full
- Interrupt generation
- DMA R/W operation
- Both 1-bit and 4-bit SD modes are supported
- SD IO card interrupt detection

20.2 Block Diagram



20.3 Command Line Control

There are three major types of responses SD card will send. They are no response, 6-byte response and 17-byte response. Programmers should set a specific response type via 0x79D2 to inform CPU how long the controller will receive the response. Following is the description of each kind of responses.

Response type 3'b000 (R0): No response.

Response type 3'b001 (R1): Normal 6-byte long response.

Response type 3'b010 (R2): 17-byte response type.

Response type 3'b011 (R3): 6-byte response with command index and 6'b111111 of CRC7 field.

Response type 3'b111 (R1b): Normal 6-byte response with a busy signal on the DAT0. Controller keeps the clock running until the busy signal is cleared.

The value of these response types may not totally appear in the response control register.

For Response type R1: only 4 bytes of bit [39:8] (card status) can be read from response control register, it has to read P_SD_RespL and P_SD_RespH for one time to get the 32-bit response.

For Response type R2: only 16 bytes of bit [127:0] (CID or CSD) can be read from response control register, it has to read P_SD_RespL and P_SD_RespH for four times to get the 128-bit response.

For Response type R3: only 4 bytes of bit [39:8] (OCR) can be read from response control register, it has to read P_SD_RespL and P_SD_RespH for one time to get the 32 bits response.

For Response type R6: only 4 bytes of bit [39:8] (RCA and Card Status) can be read from response control register, it has to read P_SD_RespL and P_SD_RespH for one time to get the 32 bits response.

In order to receive all 16 bytes (except the first byte) response via the response register, a host needs to poll the CMDBUFFULL register and read 4 bytes one by one. There are two 32-bit buffers (P_SD_RespL and P_SD_RespH) to facilitate receiving responses. The CMDBUFFULL will be set when one of two buffers is full. If both of them are full, the controller will stop the clock then wait until the host read the response register.

Responses will time out after 64 clocks cycle when the host transfer the last bit of a command. If the card do not response in this period, TIMEOUT bit will be set.

20.4 Data Line Control

If a command will have data transferred on the data line, the host needs to set the P_SD_CMD.bit8 to 1 and set P_SD_CMD.bit9 to indicate the data direction. The BLKLEN is also necessary for the controller to determine how many bytes need to be transferred.

Another thing need to note is setting the data length. The data length can be given by byte, but it is necessary to align data length in word to prevent data from losing.

In transmitting mode, the host will start to transmit extra 2 clocks after the final bit of the response. After all bits and CRC16 are transmitted, the host will wait for 2 clocks and start to receive the CRC status from the card. If the CRC status indicates the CRC fails, DATCRCERR bit will be set.

In receiving mode, the host will start to wait for data after the final bit of a command is sent. This wait will be timeout after 150ms. TIMEOUT bit will be set in such a condition. If the card transmits the data and CRC16 correctly, the transaction completes smoothly, or else the P_SD_Status.bit10 (Data CRC Error) will be set.

20.5 Card Insertion Detection

When both the CMD and DAT State machine are idle and DAT3 on the bus is pulled high, the P_SD_Status.bit12 (Card Present) will be set. Otherwise, if the DAT3 is pulled low, the P_SD_Status.bit12 (Card Present) bit will be cleared. A de-bounce circuit is used here to prevent the noise on the bus.

20.6 Multi-Block Read/Write

The multi-block read/write mode is enabled by setting P_SD_CMD.bit10 (Multi-Block Transfer) to 1. In this mode, host can read/ write multi-block in one command. The read/write method is the same as single block mode. The only difference is that the host needs to stop controller manually by setting STPCMD to 1 when all data are received/ transmitted. Host also should initiate CMD12 on the bus to stop the card.

20.7 SD/MMC Control Pin Configuration

Name	I/O	Description
SDCMD	I/O	Command / Response transfer on this pin (Shared with PortC5)
SDData0	I/O	Data Transfer Pin (Shared with PortC6)
SDData1	I/O	Data Transfer Pin (Shared with PortC7)
SDData2	I/O	Data Transfer Pin (Shared with PortC8)
SDData3	I/O	Data Transfer Pin (Shared with PortC9)
SDCLK	O	Clock Pin (Shared with PortC4)

20.8 Control Register

SD/MMC Register Summary Table

Name	Address	Description
P_SD_DataTX	0x79D0	SD/MMC Data Transmit Register
P_SD_DataRX	0x79D1	SD/MMC Data Receive Register
P_SD_CMD	0x79D2	SD/MMC Command Register
P_SD_ArgL	0x79D3	SD/MMC Argument Low Word Register
P_SD_ArgH	0x79D4	SD/MMC Argument High Word Register
P_SD_Respl	0x79D5	SD/MMC Response Low Word Register
P_SD_Resph	0x79D6	SD/MMC Response High Word Register
P_SD_Status	0x79D7	SD/MMC Status Register
P_SD_Ctrl	0x79D8	SD/MMC Control Register
P_SD_BLKLEN	0x79D9	SD/MMC Block Length Register
P_SD_INT	0x79DA	SD/MMC Interrupt Enable Register

P_SD_DataTX		0x79D0								SD/MMC Data Transmit Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DataTX																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Function	Type	Description	Condition
[15:0]	DataTX	R/W	SD/MMC Data Transmit Register Write data to SD card, data can be written to this register only when DATBUFEMPTY is 1.	

Data transmit register, host writes 16-bit data to this register and the controller will transmit it to SD card. When the data stored in the buffer is transmitted, **DATBUFEMPTY** bit in **P_SD_Status** register will be set or the DMA request will be issued. It should be noted data could be written to this register only when **DATBUFEMPTY** is 1.

P_SD_DataRX		0x79D1							SD/MMC Data Receive Register								
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DataRX															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	DataRX	R/W	SD/MMC Data Receive Register Read data from SD card, read data from this register will only valid when the DATBUFFULL is set, otherwise, it will return zeros.	

Data receive register. This register is used to store the data read from the SD card. When 16-bit data is received, DATBUFFULL bit in status register will be set or the DMA request will be issued. It should be noted data could be read from this register only when **DATBUFFULL** is 1.

P_SD_CMD					0x79D2			SD/MMC Command Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	RespType			IniCard	MulBlk	TranData	CmdWD	RunCmd	StpCmd	CmdCode					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
15			Reserved	
[14:12]	RespType	R/W	Response Type Selection. Indicate the response type of this command. Currently, only the response type R2 has response length 128 bits, all other response will have 32 bits in length. Response type R1b will keep the controller to wait for busy signal on the SD bus.	000= No response. 001= Response type R1. 010= Response type R2. 011= Response type R3. 110= Response type R6. 111= Response type R1b.
11	IniCard	R/W	Initial Card Write This Bit to 1 will initiate 74 clock cycles on the clock line.	
10	MulBlk	R/W	Multi-Block transfer bit. If this bit is set to "1", it will initiate a multiple block transfer.	0= Single block transfer. 1= Multiple block transfer.
9	TranData	R/W	Transmit / Receive Data. Indicate if this command transmits or receives data.	0= Receive data. (Read) 1= Transfer data. (Write)
8	CmdWD	R/W	Command With Data. Indicate if this command is with or without data.	0= Command without data. 1= Command with data.
7	RunCmd	R/W	Run Command. Write '1' to this register will initiate the SD command on the SD bus according to current configuration of the controller. This bit will be cleared to '0' after the transaction starts. You can start a new transaction only when BUSY bit is 0.	
6	StpCmd	R/W	Stop Command. Write '1' to this bit will force the controller back to IDLE state. This bit will be clear to '0' after the controller is back to IDLE state.	

Bit	Function	Type	Description	Condition
[5:0]	CmdCode	R/W	Command Code. The command code is that host wishes to transfer.	

P_SD_ArgL 0x79D3 SD/MMC Argument Low Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ArguMentL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ArguMentL	R/W	Argument [15:0] transfer to SD card.	

P_SD_ArgH 0x79D4 SD/MMC Argument High Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ArguMentH															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	ArguMentH	R/W	Argument [31:16] transfer to SD card.	

Host writes the argument to be transferred to the card in this register. The SD command needs a 32-bit command. Host must fill ARGUMENTL first, and then fill ARGUMENTH next to ensure a 32-bit command is transmit correctly.

P_SD_RespL 0x79D5 SD/MMC Response Low Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RespL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	RespL	R	Response Data from SD card. Reading data from this register will be valid only if the CMDBUFFULL is set.	

P_SD_RespH 0x79D6 SD/MMC Response High Word Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RespH															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	RespH	R	Response Data from SD card. Reading data from this register will be valid only if the CMDBUFFFULL is set.	

This register is used to store the response from the SD card. Commands with response R1, R1b, R3, or R6 have 6-bit command index and 32-bit response length. The response will be stored in this register. Commands with response R2 have response length in 128 bits. Host needs to poll the **CMDBUFFFULL** bit in the status register to determine when to read this register. The data in this register is valid only when **CMDBUFFFULL** bit is '1'. Host must read P_SD_RespL first and then read P_DS_RespH next to ensure every 32-bit response is received correctly. For response R2, it has to read P_DS_RespL and P_DS_RespH for four times to get the 128-bit response.

P_SD_Status			0x79D7					SD/MMC Status Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	CINT	CPRE/C	CWP	DCRCE/C	TO/C	DBufEpt	DBuffFu	RBufFu	RCRCE/C	RidxE/C	DCOM/C	CCOM/C	CBY	BY
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:14]			Reserved	
13	CINT	R	Card Interrupt Indicate a SD IO card interrupt is pending. This bit will be set only when IOEN in control register is 1. Host needs to clear the interrupt by using device specific command. Write 1 to this register will have no effect.	Read 0= Not occurred Read 1= Occurred
12	CPRE/C	R/W	Card Present This bit is only to detect the DAT3 on the SD interface when the controller is idle. Controller's behavior will not be affected by this bit. Host can initiate a transaction no matter what this bit is. Writing 1 to this register will clear the pending interrupt of card present.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear
11	CWP	R	Card Write Protect Indicate the card is writing-protect. This bit is only to detect the writing-protect pin on the interface. Controller's behavior will not be affected by this bit. Host is responsible for protecting the card.	Read 0= Not protect Read 1= Protect
10	DCRCE/C	R/W	Data CRC Error Indicate read data CRC error or write data with CRC error response.	Read 0= Not error Read 1= Error Write 0= No effect Write 1= Clear

Bit	Function	Type	Description	Condition
9	TO/C	R/W	Time Out Indicate command response time out or read data response time out.	Read 0= Not timeout Read 1= Timeout Write 0= No effect Write 1= Clear
8	DBufEpt	R	Data Buffer Empty This bit will be set when data buffer is empty. This bit will be cleared after data had been written to the DATATx register or after writing 1 to StpCmd bit in P_SD_CMD.	Read 0= Buffer Not Empty Read 1= Buffer Empty
7	DBufFu	R	Data Buffer Full This bit will be set when data buffer is full. This bit will be clear after data had been read from the DATARx register or after writing 1 to StpCmd bit in P_SD_CMD.	Read 0= Buffer Not Full Read 1= Buffer Full
6	RBufFu	R	Response Buffer Full Indicate the RESP register is full. Reading data from RESP register or initiating a new transaction or setting STPCMD in command register will clear this bit.	Read 0= Buffer Not Full Read 1= Buffer Full
5	RRCRC/C	R/W	Response CRC Error Indicate the CRC bits in the response are failed. This bit will be set if the CRC received is not 6'b111111 in the case of response R3.	Read 0= Not error Read 1= Error Write 0= No effect Write 1= Clear
4	RidxE/C	R/W	Command Index in Response Error Indicate the command index in the response is failed.	Read 0= Not error Read 1= Error Write 0= No effect Write 1= Clear
3	DCOM/C	R/W	Data Complete Indicate data transmitting/receiving is complete.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear
2	CCOM/C	R/W	Command Complete Indicate corresponding response is received or a timeout happens after sending a command.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear
1	CBY	R	SD Card Busy Indicate the SD card is busy (drive the DAT0 low). Host needs to poll this bit after a "Write" command is issued.	0 = Card is not Busy. 1 = Card is Busy.
0	BY	R	Controller busy Indicate the controller is busy.	0 = Controller is idle. 1 = Controller is busy.

P_SD_Ctrl				0x79D8				SD/MMC Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	SDEN	IOEN	DMAMOD	BUSWD	CLKDIV							
Default	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
11	SDEN	R/W	SD Enablen If this bit is set to "1", SD/MMS interface is enabled. Or, SD/MMC interface is disabled.	0= Disable 1= Enable
10	IOEN	R/W	SD IO Card Interrupt Enable If this bit is set to "1", SD IO Card interrupt detection is enabled; else it is disabled.	0= Disable 1= Enable
9	DMAMOD	R/W	DMA Mode Enable If this bit is set to "1", it will use DMA channel to transfer data.	0= Not using DMA mode 1= Using DMA mode
8	BUSWD	R/W	Bus Width Selection If this bit is set to "1", the data bus width is 4 bits during a transfer; else the bus width is 1 bit.	0= 1 bit data bus 1= 4 bits data bus
[7:0]	CLKDIV	R/W	Clock Division The clock speed on the SD bus is calculated from these bits. $FSDCLK = FSYSCLK/2(CLKDIV+1)$	

SD/MMC control register is used to control the clock speed of the SD bus and data block length when transmitting or receiving data. This register is changeable only when BUSY bit in status register is '0'.

P_SD_BLKLEN				0x79D9				SD/MMC Block Length Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	BLKLEN											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:12]			Reserved	
[11:0]	BLKLEN	R/W	Data Block Length The data block length to be transferred is in the unit of bytes. The value in this register should be equal to the block length of the SD/MMC card.	

P_SD_INT
0x79DA
SD/MMC Interrupt Enable Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	IOINT	INSINT	DBULEPT	DBULFU	CBULFU	DCOM	CCOM
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:7]			Reserved	
6	IOINT	R/W	SD IO Card Interrupt Enable Writing "1" to this bit will enable the SD IO card interrupt.	0= Disable 1= Enable
5	INSINT	R/W	Card Insert Interrupt Enable Writing "1" to this bit will enable the card insert interrupt. Writing "1" to the P_SDStatus [12] will clear this interrupt.	0= Disable 1= Enable
4	DBULEPT	R/W	Data Buffer Empty Interrupt Enable Writing "1" to this bit will enable the data buffer empty interrupt. This interrupt will be cleared after data had been written to the P_SD_DataTX.	0= Disable 1= Enable
3	DBULFU	R/W	Data Buffer Full Interrupt Enable Writing "1" to this bit will enable the data buffer full interrupt. This interrupt will be cleared after data had been read from the P_SD_DataTR.	0= Disable 1= Enable
2	CBULFU	R/W	Command Buffer Full Interrupt Enable Writing "1" to this bit will enable the command buffer full interrupt. This interrupt will be cleared when read from P_SD_Resp register or start a new transaction or set STPCMD in command register.	0= Disable 1= Enable
1	DCOM	R/W	Data Complete Interrupt Enable Write "1" to this bit will enable the data complete interrupt. Writing "1" to P_SD_Status [3] will clear this interrupt.	0= Disable 1= Enable
0	CCOM	R/W	Command Complete Interrupt Enable Write "1" to this bit will enable the command complete interrupt. Writing "1" to P_SD_Status [2] will clear this interrupt.	0= Disable 1= Enable

20.9 Example Program

Read_SD_Sector:

```
[P_SD_ArgL]=r3
[P_SD_ArgH]=r4
r2=0x1191           // Send SDC Command17, Response R1
[P_SD_CMD]=r2       // SDC Command Run, SDC Command
                    // with data

call Wait_CMD_Complete

r1=Response          // receive command response
r2=[P_SD_RespL]
[r1++]=r2
r2=[P_SD_RespH]
[r1]=r2

r1=Store_High_Addr   // store High word
r2=Store_Low_Addr    // store Low word
ds=r1
r4=256               // Read 256 words
```

Read_Finish?:

```
call Wait_Data_Full
r3=[P_SD_DataRx]
ds:[r2++]=r3
r4-=1
jnz Read_Finish?

call Wait_Controller_Busy
call Wait_Data_Complete
call Wait_Card_Busy
.....
```

Write_SD_Sector:

```
[P_SD_ArgL]=r3
[P_SD_ArgH]=r4
r2=0x1398           // Send SDC Command 24, Response R1,
[P_SD_CMD]=r2       // SDC Command Run, SDC Command
call Wait_CMD_Complete // with Data, SDC Transmit Data

r1=Response          // Receive response
r2=[P_SD_RespL]
[r1++]=r2
r2=[P_SD_RespH]
[r1]=r2
```

```

r1=Load_High_Addr      // Load High word
r2=Load_Low_Addr       // Load Low word
ds=r1
r4=256                  // Write 256 words

Write_Finish?:

    call Wait_Data_Empty
    r3=ds:[r2++]
    [P_SDC_DataTx]=r3
    r4-=1
    jnz Write_Finish?

    call Wait_Controller_Busy
    call Wait_Data_Complete
    call Wait_Card_Busy
```

21 Key Scan Controller

21.1 Introduction

The key-scan controller of GPL162002A/162003A provides the hardware key-scan function which shares IO with LCD interface without affecting the LCD display. When LCD is not turned on, the key-scan controller can still work. The key-scan controller supports up to 64 keypads when using IOA [15:0]. Hardware interrupt and auto-detect function are also provided.

- Support up to 64 keypads. Share IO with LCD interface
- Key-scan function still works when LCD is off or IOA is not shared
- Interrupt generation
- Automatic detection of key being pressed
- Support inverted output
- Automatic sample mode
- Manual sample mode
- Programmable sampling time

21.2 Key Scan Function

The key scan function of GPL162002A/162003A configures IOA as output state at IOA [7:0] and input state at IOA [15:8]. The IOA [7:0] can be shared with LCD data [7:0], which means the key scan function can still work even when LCD data is used on IOA [7:0]. It should be noted that when key scan function shares IOA with LCD panel, each output must connect a diode serially to the key pad to prevent the LCD glitch caused by contention when multi-keys are pressed. The following diagram shows an example of the connection.

If users use TFT LCD as parallel mode (16-bit data transmission), IOA [15:0] are all at output state. So the keyscan function on this mode is invalid.

21.3 Key Scan Application Circuit

Before using the key-scan function, users must set IOA [15:8] to input mode. If not all of input pins are used, set necessary pins of IOA to input mode is still acceptable. But the auto-detect interrupt may work incorrectly if the unused IOA is toggled when the scan is processing. Figure 1 and 2 shows the application circuit for key scan function.

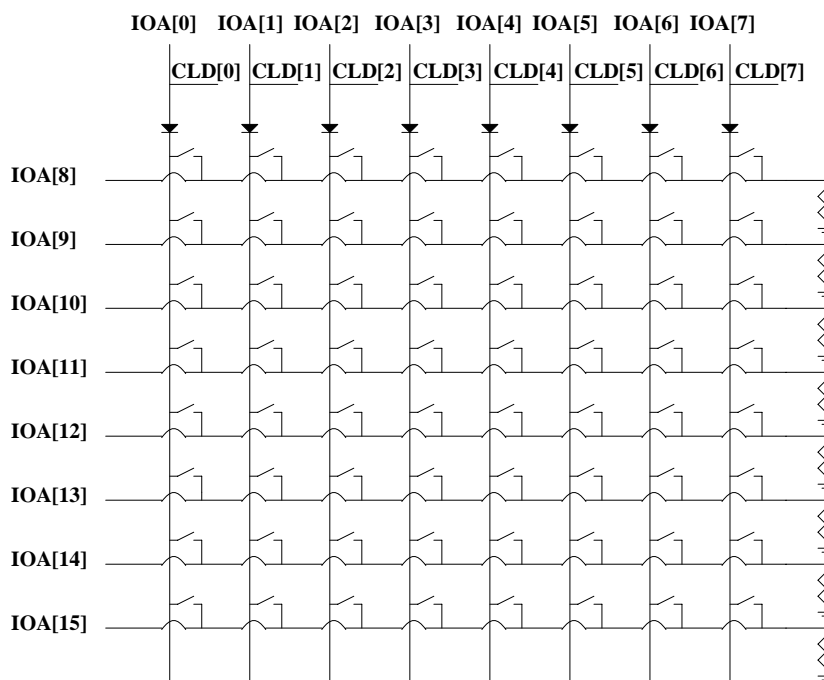


Fig1. Key Scan Application Circuit when INV in P_KS_Ctrl is set to 0.

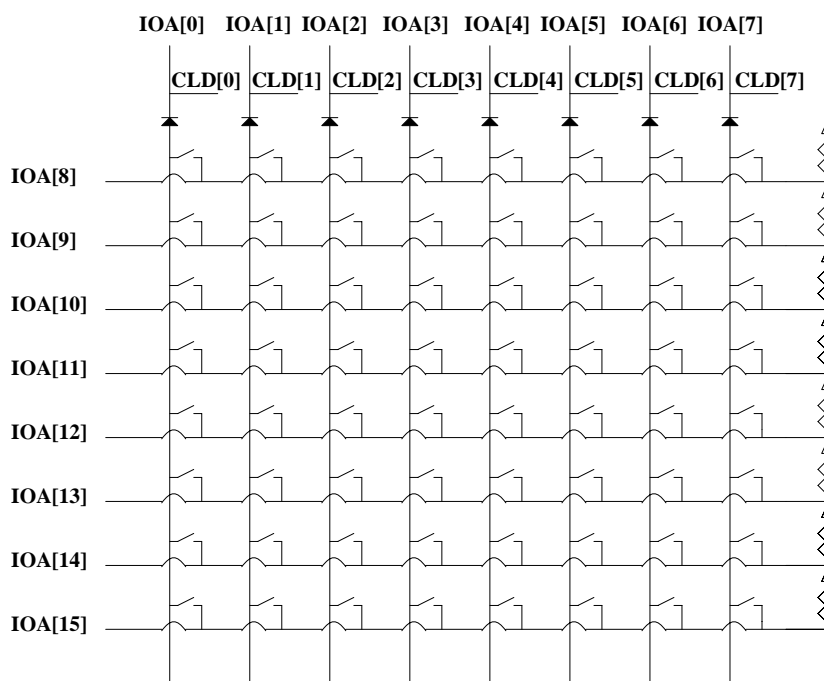


Fig2. Key Scan Application Circuit when INV in P_KS_Ctrl is set to 1.

21.4 Sample Time Configuration

There are two modes of the sampling time of the key scan controller. One is used only when the LCD is turned on and shares IO with key scan controller. In this mode, the scan time depends on the blank time of LCD interface between each line; the length of this period is 6 T at least and 100 T at most depending on users' definition on the LCD controller. All the blank time will be used as one sampling time; consequently, it will take a sampling time to complete an 8-line key-scan process.

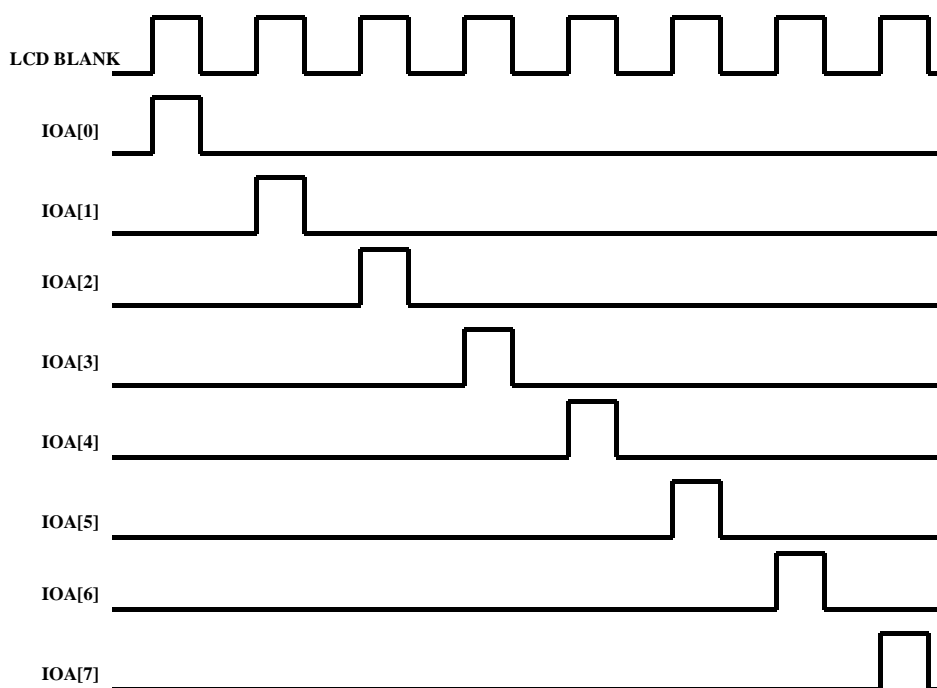


Fig3. Key Scan Mode 1, Flexible Sampling time depends on the LCD's configuration.

The other type is the fixed sampling time mode. This mode is executed automatically when LCD is off or LCD does not share IO with key scan controller. Writing FIXSTIME in P_KS_Ctrl to 1 can enter this mode when this controller shares IO with the LCD interface. The sampling time can be configured as 8T, 16T, 32T, and 64T. When the resistance or the capacitance connected with a key pad is larger, using longer sampling time will have a better result.

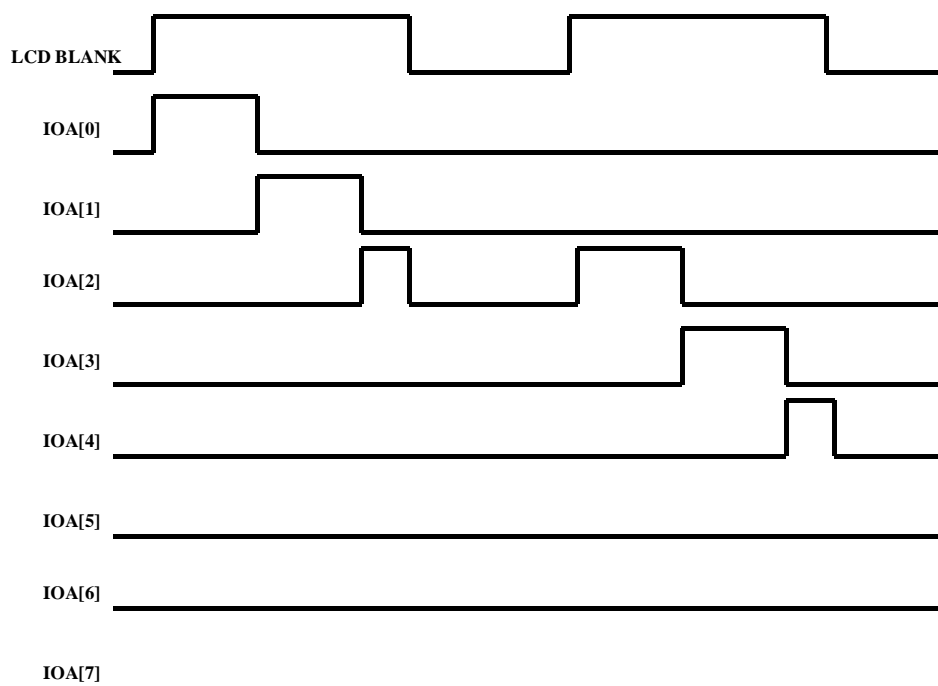


Fig4. Key Scan Mode 2, Fixed Sampling time depends on the TSEL in P_KS_Ctrl; the terminated key-scan will continue at next LCD blank.

21.5 Auto / Manual Sample Mode

The key scan controller provides auto sample mode and manual mode. The auto sample mode is initiated by writing 1 to AUTO in P_KS_Ctrl. Each time the selected timer overflow bit is rising, the key scan function will start automatically. After the key scan process is complete, the INT in P_KS_Ctrl will be set (only when SMART in P_KS_Ctrl is 0 or when SMART is 1 and a key is pressed). If IEN in P_KS_Ctrl is 1, IRQ6 or FIQ will be triggered depending on the setting of P_INT_Priority2.

The manual mode is similar to auto mode, except the trigger source needs to write '1' to STRSCAN in P_KS_Ctrl register. The scan process is the the same as the auto mode. It should be noted when the BY in P_KS_Ctrl is 1, any trigger event, including the manual trigger or the auto trigger, is invalid and will be ignored.

21.6 Automatically Detect Key Process

The key scan controller can automatically detect if any key is pressed after a key scan process is complete. To enable this function, programmers should write '1' to SMART in P_KS_Ctrl. After the function is enabled, the INT in P_KS_Ctrl will be set only when a key is pressed. This will reduce the firmware loading for key scan function.

21.7 Key Scan Control Pin Configuration

Name	I/O	Description
Keyout[7:0]	O	The keyout is through IOA[7:0]
Keyin[7:0]	I	The keyin is through IOA[15:8]

21.8 Control Register

Key Scan Register Summary Table

Name	Address	Description
P_KS_Ctrl	0x7BC0	Key Scan Control Register
P_KS_Data0	0x7BC8	Sample Data of Line IOA[0]
P_KS_Data1	0x7BC9	Sample Data of Line IOA[1]
P_KS_Data2	0x7BCA	Sample Data of Line IOA[2]
P_KS_Data3	0x7BCB	Sample Data of Line IOA[3]
P_KS_Data4	0x7BCC	Sample Data of Line IOA[4]
P_KS_Data5	0x7BCD	Sample Data of Line IOA[5]
P_KS_Data6	0x7BCE	Sample Data of Line IOA[6]
P_KS_Data7	0x7BCF	Sample Data of Line IOA[7]

P_KS_Ctrl		0x7BC0										Key Scan Control Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
Function		INT/C	IEN	AUTO	FIXSTIME	INV	SMART	STRSCAN	BY	STOP	B74OFF	B31OFF	B0OFF	STIME		TSEL					
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				

Bit	Function	Type	Description	Condition
15	INT	R/W	Key Scan Interrupt Flag This bit is set to "1" by hardware if the key scan interrupt happens.	Read 0= Not occurred Read 1= Occurred Write 0= No effect Write 1= Clear the flag
14	IEN	R/W	Key Scan Interrupt Enable If this bit is set to "1" and key scan interrupt occurs, hardware will issue and IRQ6 or FIQ to CPU. If this bit is cleared to "0", this interrupt will be masked. To select between IRQ6 and FIQ, please refer to Chapter Interrupt .	0= Disable 1= Enable
13	AUTO	R/W	Automatically Sample Mode When this bit is set to "1", the key scan controller will initiate a scan process automatically when selected timer overflow happens.	0= Manual scan mode only 1= Automatic and manual scan mode are both available.

Bit	Function	Type	Description	Condition
12	FIXSTIME	R/W	Fix Sample Time When LCD is turned off, this bit is useless and the controller will change to fixed sampling time mode automatically. When LCD is on and uses IOA as output, users can choose either LCD blank time or fixed sampling time as sampling time via this bit.	0= Use LCD blank time as sample time. 1= Use fixed sampling time set in STIME as sampling time.
11	INV	R/W	Inverted Output Control The key scan controller supports pull-up circuit or pull-down circuit for external keypads.	0= External keypads with pull-down circuit. The scan output is high active. 1= External keypads with pull-up circuit. The scan output is low active.
10	SMART	R/W	Automatically Detect Key Press Mode The key scan controller can detect if any key is pressed in a scan.	0= Interrupt is asserted each time when keyscan is complete. 1= Interrupt is asserted each time when keyscan is complete and a key is pressed.
9	STRSCAN	W	Manual Start a Key-Scan. When users wish to initiate a key scan manually, writing '1' to this bit will generate a key-scan procedure. But if the controller is busy when users write 1 to this register, the request will be ignored. Programmers must make sure the controller is not busy before writing 1 to this register.	Write 0=No effect Write 1=Start a Key Scan process.
8	BY	R	Key Scan Controller Busy Status This bit indicates if the key scan controller is idle or busy.	0= idle 1= Busy
7	STOP	W	Force to Stop the Key Scan Controller When key scan contrller works incorrectly, programmers can write 1 to this bit to force the key scan controller to stop. This will make the key scan controller back to idle state.	Write 0= No effect Write 1= Stop the controller
6	B74OFF	R/W	Bit [7:4] OFF When this bit is set to 1, IOA7~IOA4 are used as GPIOs, and setting the control registers, P_KS_Data7~P_KS_Data4, is invalid.	Write 0= IOA7~IOA4 are used as scan output. Write 1= IOA7~IOA4 are used as GPIOs.

Bit	Function	Type	Description	Condition
5	B31OFF	R/W	Bit [3:1] OFF When this bit is set to 1, IOA3~IOA1 are used as GPIOs, and setting the control registers, P_KS_Data3~P_KS_Data1, is invalid.	Write 0= IOA3~IOA1 are used as scan output. Write 1= IOA3~IOA1 are used as GPIOs.
4	B0OFF	R/W	Bit 0 OFF When this bit is set to 1, IOA0 is used as GPIO, and setting the control register of P_KS_Data0 is invalid.	Write 0= IOA0 is used as scan output. Write 1= IOA0 is used as GPIOs.
[3:2]	STIME	R/W	Key Scan Sample Timer Selection These bits are valid only when FIXSTIME is set to "1" or LCD is turned off. When LCD is using IOA and FIXSTIME is set to 0, these bits is invalid, the sampling time in such a case is the length of LCD blank time.	Key scan sample time: 00= 8 System Clocks 01= 16 System Clocks 10= 32 System Clocks 11= 64 System Clocks
[1:0]	TSEL	R/W	Key Scan Auto Sample Mode Timer Selection These bits are valid only when auto-sample mode is enabled.	00= Trigger Source is TimerC 01= Trigger Source is TimerD 10= Trigger Source is TimerE 11= Trigger Source is TimerF

P_KS_Data0
0x7BC8
Sample Data of Line IOA[0]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data0	R	Scan data is read from IOA [15:8] when IOA [0] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00000001 (KSINV == 0) or 0b11111110 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data1
0x7BC9
Sample Data of Line IOA[1]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	

Bit	Function	Type	Description	Condition
[7:0]	Data1	R	Scan data is read from IOA [15:8] when IOA [1] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00000010 (KSINV == 0) or 0b11111101 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data2
0x7BCA
Sample Data of Line IOA[2]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data2							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data2	R	Scan data is read from IOA [15:8] when IOA [2] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00000100 (KSINV == 0) or 0b11111011 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data3
0x7BCB
Sample Data of Line IOA[3]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data3							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data3	R	Scan data is read from IOA [15:8] when IOA [3] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00001000 (KSINV == 0) or 0b11110111 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data4
0x7BCC
Sample Data of Line IOA[4]

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data4							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data4	R	Scan data is read from IOA [15:8] when IOA [4] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00010000 (KSINV == 0) or 0b11101111 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data5		0x7BCD								Sample Data of Line IOA[5]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data5							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data5	R	Scan data is read from IOA [15:8] when IOA [5] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b00100000 (KSINV == 0) or 0b11011111 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data6		0x7BCE								Sample Data of Line IOA[6]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data6							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data6	R	Scan data is read from IOA [15:8] when IOA [6] is active.	

This register stores the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b01000000 (KSINV == 0) or 0b10111111 (KSINV == 1). The data will be preserved until the next scan process happens.

P_KS_Data7		0x7BCF								Sample Data of Line IOA[7]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data7							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:8]			Reserved	
[7:0]	Data7	R	Scan data is read from IOA [15:8] when IOA [7] is active.	

This register store the scan data read from IOA [15:8] when the IOA [7:0] is set as 0b10000000 (KSINV == 0) or 0b01111111 (KSINV == 1). The data will be preserved until the next scan process happens.

21.9 Example Program

Manual FixSample:

```

r1=0x10000-(32768/60)           // TimerD 60Hz
[P_TimerD_Preload]=r1
r1=0xe062                       // Enable TimerD
[P_TimerD_Ctrl]=r1

r1=0x9000                       // Manual scan mode, Fix sample time
[P_KS_Ctrl]=r1

keyloop?:
r1=[P_KS_Ctrl]
jpl keyloop?
r1=[P_KS_Ctrl]                 //clear flag
[P_KS_Ctrl]=r1
r1=[P_IOB_Buffer]
r1 ^= 0x0001
[P_IOB_Buffer]=r1
r4=8
r1=0
r2=P_KS_Data0

loop?:
r3=[r2++]
jnz keynum?
r1+=8
r4-=1
jnz loop?
jmp keyloop?

keynum?:
r3=r3 lsr 1
jz keysend?
r1+=1
jmp keynum?

keysend?:
cmp r1,[_R_PreKey]
je keyloop?
[_R_PreKey]=r1
[_R_Press]=r1
jmp keyloop?

```

_IRQ4:

```
push r1 to [sp]
r1=[P_INT_Status2]
test r1,C_INT_TimerD
jz L_EndIRQ4
r1=[P_TimerD_Ctrl]
[P_TimerD_Ctrl]=r1
r1=[P_KS_Ctrl]
test r1,0x0100
jnz L_EndIRQ4
r1|=0x0200 // Write one to start a key scan process
[P_KS_Ctrl]=r1
```

L_EndIRQ4:

```
pop r1 from [sp]
reti
```


22 Miscellaneous

22.1 Introduction

For facilitating programming, GPL162002A/162003A offers bit, nibble, and byte swap operations. It can save CPU resource on these operations.

22.2 Specified Register

Miscellaneous Register Summary Table

Name	Address	Description
P_Byte_Swap	0x7BD0	Byte Swap
P_Nibble_Swap	0x7BD1	Nibble Swap
P_TwoBit_Swap	0x7BD2	Two-Bit Swap
P_Bit_Reverse	0x7BD3	Bit Reverse

P_Byte_Swap		0x7BD0						Byte Swap								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BYTESWAP	R/W	Byte Swap Write [B15:B0] to this control register and then read this control register to obtain [B7:B0, B15:B8].	

P_Nibble_Swap				0x7BD1				Nibble Swap								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	NIBSWAP	R/W	Nibble swap Write [B15:B0] to this control register and then read this control register to obtain [B11:B8, B15:B12, B3:B0, B7:B4].	

P_TwoBit_Swap				0x7BD2				2-Bit Swap								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	2BSWAP	R/W	Two bits swap Write [B15:B0] to this control register and then read this control register to get [B13:B12, B15:B14, B9:B8, B11:B10, B5:B4, B7:B6, B1:B0, B3:B2].	

P_Bit_Reverse			0x7BD3								Bit Reverse					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Function	Type	Description	Condition
[15:0]	BITREV	R/W	Bit reverse Write [B15:B0] to this control register and then read this control register to get [B0:B15].	

22.3 Program example

```

r1 = 0xFA50
[P_Byte_Swap]=r1
[P_Nibble_Swap]=r1
[P_Bit_Reverse]=r1
r1 = 0x6996
[P_TwoBit_Swap]=r1

r1=[P_Byte_Swap]
cmp r1,0x50FA // Verify Byte Swap
jne L_OtherTestError?
r2=[P_Nibble_Swap]
cmp r2,0xAF05 // Verify Nibble Swap
jne L_OtherTestError?
r3=[P_Bit_Reverse]
cmp r3,0x0A5F // Verify Bit Reverse
jne L_OtherTestError?

r4=[P_TwoBit_Swap]
cmp r4,0x9669 // Very 2-bit Reverse
jne L_OtherTestError?
.....

L_OtherTestError?:
.....

```

23 E-Fuse Option

23.1 Introduction

There are four E-Fuse registers in GPL162002A/162003A. The value of these four registers can be decided by customers on IC mass production procedure so that users can protect the program by configuring these four registers.

23.2 Specified Register

E-Fuse Register Summary Table

Name	Address	Description
P_EFuse_D0	0x7C30	E-Fuse Data Register 0
P_EFuse_D1	0x7C31	E-Fuse Data Register 1
P_EFuse_D2	0x7C32	E-Fuse Data Register 2
P_EFuse_D3	0x7C33	E-Fuse Data Register 3

P_EFuse_D0		0x7C30					E-Fuse Data Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [15:0]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:0]	E-DATA	R	E fuse data out [15:0]	

P_EFuse_D1		0x7C31						E-Fuse Data Register 1								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [31:16]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:0]	E-DATA	R	E fuse data out [31:16]	

P_EFuse_D2		0x7C32						E-Fuse Data Register 2								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [47:32]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Function	Type	Description	Condition
[15:0]	E-DATA	R	E fuse data out [47:32]	

P_EFuse_D3
0x7C33
E-Fuse Data Register 3

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [63:48]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

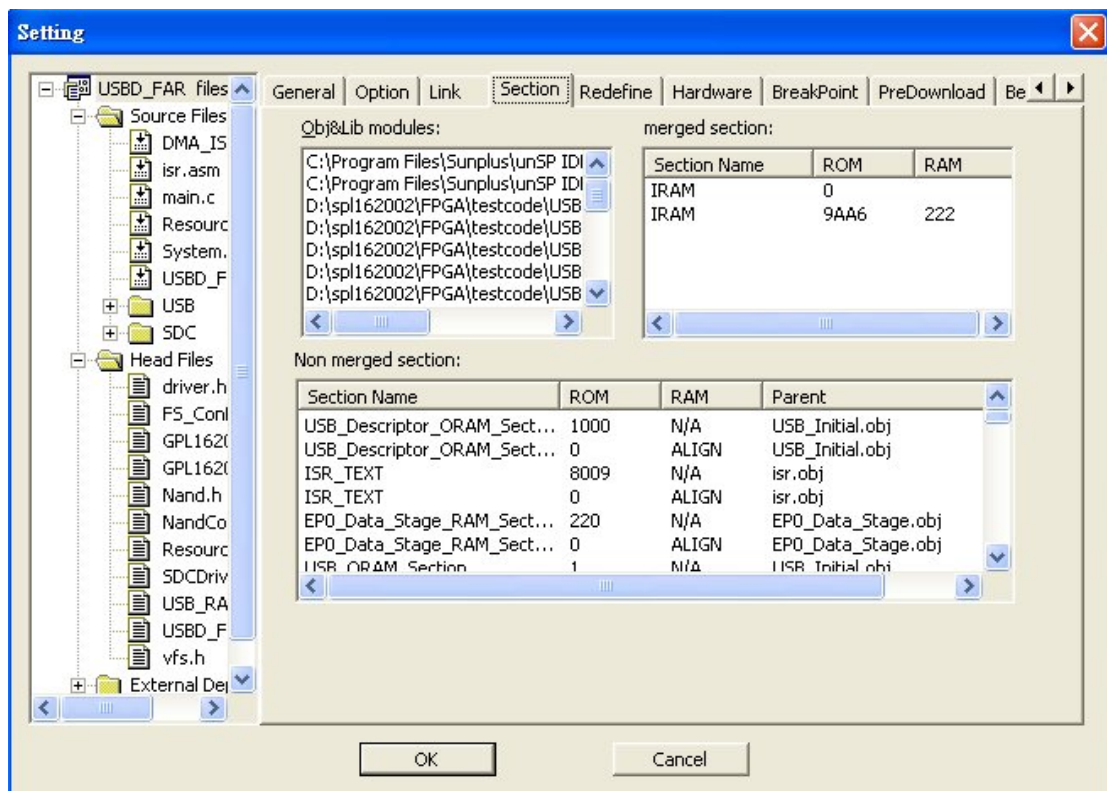
Bit	Function	Type	Description	Condition
[15:0]	E-DATA	R	E fuse data out [63:48]	

24 Link your program

24.1 Resource File Alignment by Link Script File

u'nSP IDE allocates programs and resources in two ways: fixing sections for resources and programs and modifying through the linking file(.lik). The ways to set up a start address are as follows:

project→setting...→section



Note: Since this is a "forced" setting, programmers must avoid the location overlap of programs and resources.

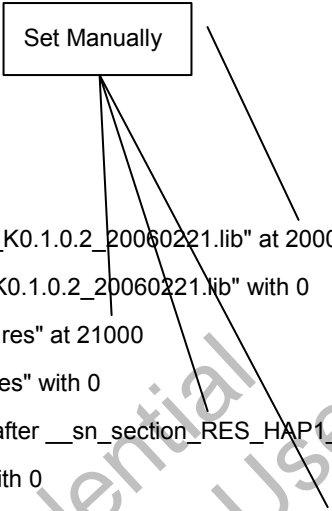
After a program is compiled and linked, a .lik file is produced. The .lik file includes the following information: obj, resource, and the definition of section link. We can allocate all resource and section location in the .lik file manually. See the following example for details:

```

;Locate: JPEG_Table in "JPEG_Table.obj" at 9000
;Align: JPEG_Table in "JPEG_Table.obj" with 0
;Locate: ICTest in "ICTest.obj" at 27000
;Align: ICTest in "ICTest.obj" with 0
;Locate: unSP_StartUp in "startup_all.obj" at FE00
;Align: unSP_StartUp in "startup_all.obj" with 0
Locate: MP3CODE_ROM in "MP3_ROM" of "MP3_HW_K0.1.0.2_20060221.lib" at 20000
;Align: MP3CODE_ROM in "MP3_ROM" of "MP3_HW_K0.1.0.2_20060221.lib" with 0
Locate: __sn_section_RES_HAP1_ADP in "HAP1_ADP.res" at 21000
;Align: __sn_section_RES_HAP1_ADP in "HAP1_ADP.res" with 0
Locate: __sn_section_RES_HB_ADP in "HB_ADP.res" after __sn_section_RES_HAP1_ADP
;Align: __sn_section_RES_HB_ADP in "HB_ADP.res" with 0
Locate: __sn_section_RES_HBQ_ADP in "HBQ_ADP.res" after __sn_section_RES_HB_ADP
;Align: __sn_section_RES_HBQ_ADP in "HBQ_ADP.res" with 0

```

Set Manually



This method is able to avoid the overlap of resources and programs when they are modified.

25 Appendix

25.1 Normally used abbreviation list

Although all abbreviation indicated here are uppercases, lower cases are also accepted for naming flexibility. However, the first letter should be capitalized. For example, "BUF" = "Buf", "CH0" = "Ch0", "CLK"="CIk", etc.

Abbreviation	Description
ADR	Address
BUF	Buffer
CHAR	Character
CHA, CHB, etc	CHA, CHB, etc.
CLK	Clock
CMP	Compare
CNT	Counter
COM	Common (starts with "0", e.g., COM0, COM1...)
DACA, DACB, etc.	DACA, DACB etc.
DEC	Decrease
DISP	Display
ECLK	External clock
ERR	Error
FG	Flag
FLOAT	Float state
FUNC	Function
INC	Increase
INT	Interrupt
DEFAULT	Initialization
LVD	Low Voltage Detection
LVPD	Low Voltage Power Down
LVR	Low Voltage Reset
MAX	Maximum
MIN	Minimum
MUTE	Mute
NMI	Non-masked-interrupt
ODN	Open drain NMOS (sink)
ODP	Open drain PMOS (send)
PortA, PortB	I/O PortA, IO PortB, etc
PH	Pull high resistor
PL	Pull low resistor
PWM	Pulse-Width Modulation
R/W	Readable/writable
ROSC	R-oscillator
RTC	Real time clock

Abbreviation	Description
SEG	Segment (starts with "0", e.g., SEG0, SEG1...)
SLP	Sleep
SPCH	Speech
STR	String
TMA, TMB, etc	TimerA, TimerB, etc.
TMP	Temporary
TONE	Tone
UART	Universal Asynchronized Receiver Transmitter
VOL	Volume
WAKE	Wakeup
WDG	Watchdog
X'TAL or XTAL	Crystal oscillator

25.2 Control Register Mapping List (by function)

0000 ₁₆ ~ 77FF ₁₆	30KW SRAM
7800 ₁₆ ~ 781F ₁₆	System Control
7820 ₁₆ ~ 785F ₁₆	Memory Control
7860 ₁₆ ~ 788F ₁₆	I/O Port Control
78A0 ₁₆ ~ 78AF ₁₆	Interrupt Control
78B0 ₁₆ ~ 78BF ₁₆	Time Base Control
78C0 ₁₆ ~ 78DF ₁₆	Timer Control
78F0 ₁₆ ~ 78FF ₁₆	Audio Output Control
7900 ₁₆ ~ 791F ₁₆	UART / IrDA Control
7920 ₁₆ ~ 793F ₁₆	RTC Control
7940 ₁₆ ~ 795F ₁₆	SPI Control
7960 ₁₆ ~ 797F ₈	Analog Control
7980 ₁₆ ~ 799F ₈	LCD Control
79A0 ₁₆ ~ 79AF ₁₆	Reserved
79B0 ₁₆ ~ 79BF ₁₆	Reserved
79C0 ₁₆ ~ 79CF ₁₆	Timer Control
79D0 ₁₆ ~ 79EF ₁₆	SD Card Interface
79F0 ₁₆ ~ 79FF ₁₆	Reserved
7A00 ₁₆ ~ 7AFF ₁₆	LCD Color Palette
7B00 ₁₆ ~ 7B2F ₁₆	USB Host
7B30 ₁₆ ~ 7B5F ₁₆	USB Device
7B60 ₁₆ ~ 7B7F ₁₆	I2C Interface
7B80 ₁₆ ~ 7BBF ₁₆	DMA Control
7BC0 ₁₆ ~ 7BCF ₁₆	Key Scan
7BD0 ₁₆ ~ 7BDF ₁₆	Miscellaneous control
7BE0 ₁₆ ~ 7BEF ₁₆	Reserved
7BF0 ₁₆ ~ 7BFF ₁₆	Audio Output Control

7C00₁₆ ~ 7C2F₁₆

7C30₁₆ ~ 7C3F₁₆

7D00₁₆ ~ 7D3F₁₆

7D40₁₆ ~ 7FFF₁₆

Reserved
E-Fuse Register
TFT Control
Reserved

System Control Register Summary Table

Name	Address	Description
P_BodyID	0x7800	Body Identification Number Register
P_CLK_Ctrl0	0x7804	Clock On/Off Control Register 0
P_CLK_Ctrl1	0x7805	Clock On/Off Control Register 1
P_Reset_Flag	0x7806	Reset Event Flag Register
P_Clock_Ctrl	0x7807	System Clock Control Register
P_LVR_Ctrl	0x7808	Low Voltage Reset Control Register
P_Watchdog_Ctrl	0x780A	Watchdog Control Register
P_Watchdog_Clear	0x780B	Watchdog Clear Register
P_WAIT	0x780C	Wait Mode Entrance Register
P_HALT	0x780D	Halt Mode Entrance Register
P_SLEEP	0x780E	Sleep Mode entrance Register
P_Power_State	0x780F	Current Power State Register
P_PLLN	0x7817	PLL's Divider selection
P_PLLWiatCLK	0x7818	PLL state change wait time
P_AD_Driving	0x781F	Address/Data Driving control Register

P_BodyID	0x7800								Body ID Number							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	0x8688															
Default	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0

P_Reset_Flag	0x7806																Reset Event Flag Register						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Function	-	-	-	-	-	-	-	-	-	-	-	WDG	WDE	MPE	-	LVR							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

P_Clock_Ctrl	0x7807																System Clock Control Register											
Bit	15	14	13	12	11	10		9	8	7	6	5	4		3	2	1	0										
Function	FAST	C32K	-	WEAK	-	C32KOFF		KCEN	-	-	-	-	DAPLLEN		CLK96M	CLKDIV												
Default	0	0	0	0	0	0		0	0	0	0	0	0		0	0												

P_CLK_Ctrl0		0x7804								Peripheral Clock Control Register0							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		Clock Source [15:0]															
Default		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

P_CLK_Ctrl1		0x7805								Peripheral Clock Control Register1							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		Clock Source [31:16]															
Default		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

P_PLLN		0x7817								Fast PLL output divider register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	PLLN						
Default		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

P_LVR_Ctrl		0x7808								Low Voltage Reset Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	-	LVROFF	-
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Watchdog_Ctrl		0x780A						Watchdog Reset Control Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WDGEN	WDGS	-	-	-	-	-	-	-	-	-	-	-	WDGPD		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Watchdog_Clear				0X780B				Watchdog Clear Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	WDGC																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_WAIT		0x780C								Wait Mode Entrance Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WAIT															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_HALT		0x780D								Halt Mode Entrance Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		HALT															
Default																	

P_Sleep		0x780E								Sleep Mode Entrance Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SLEEP															
Default																	

P_State		0x780F								Power State Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-												State			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

P_AD_Driving		0x781F								Address/Data Driving Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		D_POFF	D_PH	D_SR	D_SMT	-	D_DRIVE			-	A_SR		A_SMT	-	A_DRIVE		
Default		0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Memory Control Register Summary Table

Name	Address	Description
P_MCS0_Ctrl	0x7820	Chip Selection 0 Memory Device Control Register
P_MCS1_Ctrl	0x7821	Chip Selection 1 Memory Device Control Register
P_MCS2_Ctrl	0x7822	Chip Selection 2 Memory Device Control Register
P_MCS3_Ctrl	0x7823	Chip Selection 3 Memory Device Control Register
P_MCS4_Ctrl	0x7824	Chip Selection 4 Memory Device Control Register
P_EMUCS_Ctrl	0x7825	EMU Chip Selection Memory Device Control Register
P_MCS_Byte_Sel	0x7826	CS0~CS4 and EMUCS Word/Byte Data Select
P_MCS3_WETimingCtrl	0x7827	MCS3 WE timing control register
P_MCS4_WETimingCtrl	0x7828	MCS4 WE timing control register
P_MCS3_RDTimingCtrl	0x7829	MCS3 RD timing control register
P_MCS4_RDTimingCtrl	0x782A	MCS4 RD timing control register
P_MCS3_TimingCtrl	0x782B	MCS3 CS timing control register
P_MCS4_TimingCtrl	0x782C	MCS4 CS timing control register
P_Mem_Ctrl	0x7840	Memory Control Register
P_Addr_Ctrl	0x7841	Memory A17~A25 Control Register
P_BankSwitch_Ctrl	0x7810	Bank Switch Control Register
P_MAPSEL	0x7816	CS0 boot mapping size select register

P_MCS0_Ctrl		0x7820								CS0 Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS0SIZE								CS0MD		WARWAT		CS0WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

P_MCS1_Ctrl		0x7821								CS1 Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS1SIZE								CS1MD		WARWAT		CS1WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

P_MCS2_Ctrl		0x7822								CS2 Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS2SIZE								CS2MD		WARWAT		CS2WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

P_MCS3_Ctrl		0x7823								CS3 Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS3SIZE								CS3MD		WARWAT		CS3WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

P_MCS4_Ctrl		0x7824								CS4 Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CS4SIZE								CS4MD		WARWAT		CS4WAIT			
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

P_EMUCS_Ctrl		0x7825								EMU Device Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-								EMCMD		WARWAT		EMUCSWAIT			
Default		0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1

P_MCS_Byte_Sel		0x7826								MCS Word/Byte Data Select register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-										EMU	S4	S3	S2	S1	S0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MCS3_WETimingCtrl		0x7827								MCS3 WE timing control register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-												WEB3NUM			
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MCS4_WETimingCtrl		0x7828								MCS4 WE timing control register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-												WEB4NUM			
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MCS3_RDTimingCtrl 0x7829
MCS3 RD timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-												RDB3NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MCS4_RDTimingCtrl 0x782A
MCS4 RD timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-												RDB4NUM			
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MCS3_TimingCtrl 0x782B
MCS3 CS timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-												CSB3NUM			
Init	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

P_MCS4_TimingCtrl 0x782C
MCS4 CS timing control register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-												CSB4NUM			
Init	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

P_Mem_Ctrl 0x7840
Memory Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	WE	RD	-	MCS4	MCS3	MCS2	MCS1	MCS0
Default	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1

P_Addr_Ctrl 0x7841
Memory Address A17~A25 Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	MA23	MA22	MA21	MA20	MA19	MA18	MA17
Default	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

P_BankSwitch_Ctrl 0x7810
Bank Switch Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-											Bank				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

P_MAPSEL 0x7816
CS0 boot mapping size select register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-														MAPSEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I/O Port Control Register Summary Table

Name	Address	Description
P_IOA_Data	0x7860	I/O PortA Data Register
P_IOA_Buffer	0x7861	I/O PortA Buffer Register
P_IOA_Dir	0x7862	I/O PortA Direction Register
P_IOA_Attrib	0x7863	I/O PortA Attribution Register
P_IOB_Data	0x7868	I/O PortB Data Register
P_IOB_Buffer	0x7869	I/O PortB Buffer Register
P_IOB_Dir	0x786A	I/O PortB Direction Register
P_IOB_Attrib	0x786B	I/O PortB Attribution Register
P_IOB_Latch	0x786C	I/O PortB Latch Register for Wakeup
P_IOC_Data	0x7870	I/O PortC Data Register
P_IOC_Buffer	0x7871	I/O PortC Buffer Register
P_IOC_Dir	0x7872	I/O PortC Direction Register
P_IOC_Attrib	0x7873	I/O PortC Attribution Register
P_IOD_Data	0x7878	I/O PortD Data Register
P_IOD_Buffer	0x7879	I/O PortD Buffer Register
P_IOD_Dir	0x787A	I/O PortD Direction Register
P_IOD_Attrib	0x787B	I/O PortD Attribution Register

P_IOA_Data 0x7860 IOA Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADATA															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOA_Buffer 0x7861 IOA Buffer Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOABUF															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOA_Dir 0x7862 IOA Direction Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOADIR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOA_Attrib 0x7863 IOA Attribution Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	IOAATT															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOB_Data		0x7868					IOB Data Register										
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBDATA															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOB_Buffer		0x7869					IOB Buffer Register										
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBBUF															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOB_Dir		0x786A								IOB Direction Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBDIR															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOB_Attrib		0x786B								IOB Attribution Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOBATT															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOB_Latch		0x786C								IOB Latch for Key Change Wakeup							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	IOBLHW		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOC_Data		0x7870				IOC Data Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOCADATA															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOC_Buffer		0x7871								IOC Buffer Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOCBUF															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOC_Dir		0x7872								IOC Direction Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IOC DIR															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IOC_Attrib		0x7873																IOC Attribution Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		IOCATT																	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_IOD_Data		0x7878																IOD Data Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		IODDATA																	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_IOD_Buffer		0x7879																IOD Buffer Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		IODBUF																	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_IOD_Dir		0x787A																IOD Direction Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		IODDIR																	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_IOD_Attrib		0x787B																IOD Attribution Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		IODATT																	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Global Interrupt Control Register Summary Table

Name	Address	Description
P_INT_Status1	0x78A0	Interrupt Status Register 1
P_INT_Status2	0x78A1	Interrupt Status Register 2
P_INT_Priority1	0x78A4	Interrupt Priority Register 1
P_INT_Priority2	0x78A5	Interrupt Priority Register 2
P_MINT_Ctrl	0x78A8	Miscellaneous Interrupt Control Register

P_INT_Status1		0x78A0																Interrupt Status 1 Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		KEYIF	ADCRIF	TFTUFIF	TFTFEIF	UTIRIF	SPIIF	FPIF	TPIF	ASIF	-	AUDBIF	AUDAIF	USB	DMA	EXTBIF	EXTAIF		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_INT_Status2		0x78A1																Interrupt Status 2 Register	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		TMDIF	TMCIF	TMBIF	TMAIF	KSIF	TMBCIF	TMBBIF	TMBAIF	-	SD	I2C	NAND	-	SCHIF	ALMIF	HMSIF		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_INT_Priority1		0x78A4								Interrupt Priority 1 Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		KEYIP	ADCRIP	TFTUFIP	TFTFEIP	UTIRIP	SPIIP	FPIP	TPIP	ASPIP	-	AUDBIP	AUDAIP	USBIP	DMAIP	EXTBIP	EXTAIP
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_INT_Priority2		0x78A5								Interrupt Priority 2 Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TMDIP	TMCIP	TMBIP	TMAIP	KSIP	-	-	-	-	SD	I2C	NAND	-	-	-	-
Default		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

P_MINT_Ctrl		0x78A8								Miscellaneous Interrupt Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		KC2IF	KC2EN	KC1IF	KC1EN	KC0IF	KC0EN	-	-	-	-	EXTBIS	EXTAIS	-	-	EXTBEN	EXTAEN
Default		0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Timer Control Register Summary Table

Name	Address	Description
P_TimerA_Ctrl	0x78C0	TimerA Control Register
P_TimerA_CCP_Ctrl	0x78C1	TimerA Capture / Comparison / PWM Control Register
P_TimerA_Preload	0x78C2	TimerA Preload Register
P_TimerA_CCP_Reg	0x78C3	TimerA Capture / Comparison / PWM Register
P_TimerA_UpCount	0x78C4	TimerA up-count value
P_TimerB_Ctrl	0x78C8	TimerB Control Register
P_TimerB_CCP_Ctrl	0x78C9	TimerB Capture / Comparison / PWM Control Register
P_TimerB_Preload	0x78CA	TimerB Preload Register
P_TimerB_CCP_Reg	0x78CB	TimerB Capture / Comparison / PWM Register
P_TimerB_UpCount	0x78CC	TimerB up-count value
P_TimerC_Ctrl	0x78D0	TimerC Control Register
P_TimerC_CCP_Ctrl	0x78D1	TimerC Capture / Comparison / PWM Control Register
P_TimerC_Preload	0x78D2	TimerC Preload Register
P_TimerC_CCP_Reg	0x78D3	TimerC Capture / Comparison / PWM Register
P_TimerC_UpCount	0x78D4	TimerC up-count value
P_TimerD_Ctrl	0x78D8	TimerD Control Register
P_TimerD_Preload	0x78DA	TimerD Preload Register
P_TimerD_UpCount	0x78DC	TimerD up-count value
P_TimerE_Ctrl	0x79C0	TimerE Control Register
P_TimerE_Preload	0x79C2	TimerE Preload Register
P_TimerE_UpCount	0x79C4	TimerE up-count value
P_TimerF_Ctrl	0x79C8	TimerF Control Register
P_TimerF_Preload	0x79CA	TimerF Preload Register
P_TimerF_UpCount	0x79CC	TimerF up-count value

P_TimerA_Ctrl			0x78C0								TimerA Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAIF/C	TMAIE	TMAEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB_Ctrl			0x78C8								TimerB Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBIF/C	TMBIE	TMBEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC_Ctrl			0x78D0								TimerC Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCIF/C	TMCIE	TMCEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD_Ctrl			0x78D8								TimerD Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDIF/C	TMDIE	TMDEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerE_Ctrl			0x79C0								TimerE Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMEIF/C	TMEIE	TMEEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerF_Ctrl			0x79C8								TimerF Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMFIF/C	TMFIE	TMFEN	-	EXTASEL	EXTBSEL	-	SRCBSEL	-	SRCASEL	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerA_CCP_Ctrl			0x78C1								TimerA CCP Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPAEN	-	-	-	-	CAPASEL	-	-	CMPASEL	-	-	PWMASEL	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB_CCP_Ctrl			0x78C9								TimerB CCP Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPBEN	-	-	-	-	CAPBSEL	-	-	CMPBSEL	-	-	PWMBSEL	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC CCP_Ctrl	0x78D1															
	TimerC CCP Control Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CCPCEN		-	-	-	-	CAPCSEL		-	-	CMPCSEL		-	-	PWMCSEL	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerA Preload	0x78C2															
	TimerA Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB Preload	0x78CA															
	TimerB Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC Preload	0x78D2															
	TimerC Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD Preload	0x78DA															
	TimerD Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerE Preload	0x79C2															
	TimerE Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMEPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerF Preload	0x79CA															
	TimerF Preload Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMFPLR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerA CCP_Reg	0x78C3															
	TimerA Capture / Comparison / PWM Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMACCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB_CCP_Reg	0x78CB				TimerB Capture / Comparison / PWM Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC CCP_Reg	0x78D3					TimerC Capture / Comparison / PWM Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCCCPR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerA UpCount				0x78C4				TimerA Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMAUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerB UpCount		0x78CC						TimerB Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerC UpCount		0x78D4						TimerC Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerD UpCount				0x78DC				TimerD Up-Count								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMDUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimerE	UpCount					0x79C4						TimerE Up-Count					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TMEUCR																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TimerF UpCount	0x79CC					TimerF Up-Count										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMFUCR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Timebase Control Register Summary Table

Name	Address	Description
P_TimeBaseA_Ctrl	0x78B0	TimeBaseA Control Register
P_TimeBaseB_Ctrl	0x78B1	TimeBaseB Control Register
P_TimeBaseC_Ctrl	0x78B2	TimeBaseC Control Register
P_TimeBase_Reset	0x78B8	TimeBase Counter Reset Register

P_TimeBaseA_Ctrl			0x78B0								TimeBaseA Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBAIF/C	TMBAIE	TMBAEN	-	-	-	-	-	-	-	-	-	-	-	TMBAS	
Default	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0

P_TimeBaseB_Ctrl			0x78B1				TimeBaseB Control Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBBIF/C	TMBBIE	TMBBEN	-	-	-	-	-	-	-	-	-	-	-	TMBBS	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TimeBaseC_Ctrl			0x78B2			TimeBaseC Control Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMBCIF/C	TMBCIE	TMBCEN	-	-	-	-	-	-	-	-	-	-	-	TMBCS	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0*	0

P_TimeBase_Reset						TimeBase Reset Control Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TMCCR															
Default																

Real Time Clock Control Register Summary Table

Name	Address	Description
P_Second	0x7920	Second Register
P_Minute	0x7921	Minute Register
P_Hour	0x7922	Hour Register
P_Alarm_Second	0x7924	Alarm Second Register
P_Alarm_Minute	0x7925	Alarm Minute Register
P_Alarm_Hour	0x7926	Alarm Hour Register
P_RTC_Ctrl	0x7934	HMS / Alarm / Scheduler Control Register
P_RTC_INT_Status	0x7935	HMS / Alarm / Scheduler Interrupt Flag & Clear Register
P_RTC_INT_Ctrl	0x7936	HMS / Alarm / Scheduler Interrupt Control Register
P_RTC_HMSBusy	0x7937	RTC HMS controller busy register

P_Second	0x7920										Second Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCSEC					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Minute	0x7921										Minute Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RTCMIN					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Hour	0x7922										Hour Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	RTCHR				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Alarm_Second	0x7924										Alarm Second Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMSEC					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Alarm_Minute	0x7925										Alarm Minute Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	ALMMIN					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Alarm_Hour	0x7926										Alarm Hour Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	ALMHR				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_RTC_Ctrl	0x7934										HMS / Alarm / Schedule Control Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RTCEN	-	-	-	-	ALMEN	HMSSEN	SCHEN	-	-	-	-	-	-	SCHSEL	
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_RTC_INT_Status	0x7935										HMS/ Alarm / Schedule Interrupt Flag & Clear Register					
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	ALMIEF/C	-	SCHIF/C	-	-	-	-	HRIF/C	MINIF/C	SECIF/C	HSECIF/C
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_RTC_INT_Ctrl					0x7936										HMS / Alarm / Schedule Interrupt Control Register			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	-	-	-	-	-	ALMIEN	-	SCHIEEN	-	-	-	-	HRIEN	MINIEN	SECIEN	HSECIEN		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_RTC_HMSBusy					0x7937										RTC HMS Busy Register			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	SEC_BUSY	MIN_BUSY	HR_BUSY	-	-	-	-	-	-	-	-	-	-	-	-	-		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

DAC Control Register Summary Table

Name	Address	Description
P_CHA_Ctrl	0x78F0	Channel A Control Register
P_CHA_Data	0x78F1	Channel A Data Register
P_CHA_FIFO	0x78F2	Channel A FIFO Control Register
P_CHB_Ctrl	0x78F8	Channel B Control Register
P_CHB_Data	0x78F9	Channel B Data Register
P_CHB_FIFO	0x78FA	Channel B FIFO Control Register
P_DAC_Ctrl	0x78FD	DAC Control Register
P_HPAMP_Ctrl	0x78FE	Headphone Amplifier Control Register
P_DAC_IIS_Ctrl	0x78FF	DAC IIS Mode Control Register
P_DAC_ACCREQ	0x7BF0	3D/EQ/AC Parameter Access Request Register
P_DAC_ACCDINL	0x7BF1	3D/EQ/AC Parameter Data Input Low Register
P_DAC_ACCDINH	0x7BF2	3D/EQ/AC Parameter Data Input High Register
P_DAC_EFF_Ctrl	0x7BF3	3D/EQ/AC Control register
P_DAC_ACTHRESL	0x7BF4	AC Threshold Low register
P_DAC_ACTHRESH	0x7BF5	AC Threshold High register
P_DAC_EQBANDSEL	0x7BF6	EQ Band Index Selection
P_DAC_EQSPEC	0x7BF7	EQ Band Spectrum output
P_DAC_VOLUME3D	0x7BF8	3D Main Volume
P_DAC_VOLUME3D_C	0x7BF9	3D Center Volume
P_DAC_VOLUME3D_S	0x7BFA	3D Surround Volume
P_DAC_VOLUME3D_R	0x7BFB	3D Right Channel Volume
P_DAC_VOLUME3D_L	0x7BFC	3D Left Channel Volume
P_DAC_ACCDOUHL	0x7BFE	3D/EQ/AC Parameter Data Output Low Register
P_DAC_ACCDOUHL	0x7BFF	3D/EQ/AC Parameter Data Output High Register

P_CHA_Ctrl					0x78F0										CHA DAC/PWM Control Register			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	FEMIF/C	FEMIEN	CHAEN	-	SIGNED	SRCEN	SRCRST	-	-	-	-	-	-	SRCFS				
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_CHA_Data		0x78F1				CHA DAC/PWM Data Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CHADATA															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_CHA_FIFO		0x78F2				CHA FIFO Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		FFUL	FUDN	-	-	-	-	-	FRST	CHAFEILV				CHAFINX			
Default		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

P_CHB_Ctrl		0x78F8				CHB DAC/PWM Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		FEMIF/C	FEMIEN	CHBEN	SSF	CHACFG	MONO	-	-	-	-	-	-	-	-	-	-
Default		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_CHB_Data		0x78F9				CHB DAC/PWM Data Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		CHBDATA															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_CHB_FIFO		0x78FA				CHB FIFO Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		FFUL	FUDN	-	-	-	-	-	FRST	CHBFEILV				CHBFINX			
Default		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

P_DAC_Ctrl		0x78FD																DAC Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0								
Function		-	-	-	-	BPFIR	AS_S	SP_DLY		AS_CYCLE		AS_RANGE		PWDAL	PWDAR	IIS	DACLK								
Default		0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0								

P_HPAMP_Ctrl		0x78FE				Headphone Amplifier Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	PWSPVR	SPINS		PWSPL	PWSPR
Default		0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

P_DAC_IIS_Ctrl		0x78FF				DAC IIS Mode Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	IIS_MCLK		IIS_BITS		IIS_MODE		IISEXT	IISEN
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_ACCREQ		0x7BF0						3D/EQ/AC Parameter AccessRequest Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	RDY	WRITE	-	-	-	-	3D	ADDR								
Default	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_ACCDINL		0x7BF1					3D/EQ/AC Parameter Data Input Low Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DATAIN[15:0]															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_ACCDINH		0x7BF2						3D/EQ/AC Parameter Data Input High Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DATAIN[23:16]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_EFF_Ctrl		0x7BF3						3D/EQ/AC Control register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	3DEN	2CH	HP	EQEN	BPEQ	BPAC	-	DEPTH_3D								
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_ACTHRESL		0x7BF4					AC(anti-clip) Threshold Low Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ACTHRES[15:0]															
Default	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_ACTHRESH		0x7BF5							AC(anti-clip) Threshold High Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	ACTHRES[23:16]							
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

P_DAC_EQBANDSEL		0x7BF6		EQ Band Index Selection												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	EQBAND		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_EQSPEC				0x7BF7				EQ Band Spectrum output								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	EQSPEC											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_VOLUME3D		0x7BF8								3D Main Volume							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		VOL_3D															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_VOLUME3D_C		0x7BF9																3D Center Volume															
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																
Function		VOL_3D_C																															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																

P_DAC_VOLUME3D_S 0x7BFA					3D Surround Volume											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	VOL_3D_S															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DAC_VOLUME3D_R 0x7BFB		3D Right Channel Volume															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	VOL_3D_R																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DAC_VOLUME3D_L 0x7BFC					3D Left Channel Volume												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	VOL_3D_L																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DAC_ACCDOUTL		0x7BFE					3D/EQ/AC Parameter Data Output Low Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	DATAOUT[15:0]																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DAC_ACCDOUTH		0x7BFF								3D/EQ/AC		Parameter	Data	Output	High				
		Register																	
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function		-	-	-	-	-	-	-	-	DATAOUT[23:16]									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

STN LCD Control Register Summary Table

Name	Address	Description
P_LCD_Setup	0x7980	LCD setup register
P_LCD_Clock	0x7981	LCD clock register
P_LCD_Segment	0x7982	LCD segment number register
P_LCD_Common	0x7983	LCD common number register

Name	Address	Description
P_LCD_Buffer_LowAdr	0x7984	LCD start address register (A0~A15)
P_LCD_Buffer_HighAdr	0x7985	LCD start address register (A16~A25)
P_LCD_Buffer_Offset	0x7986	LCD virtual page offset register
P_LCD_Timing_Ctrl	0x7987	LCD control signal timing register
P_LCD_Frame_Ctrl	0x7988	LCD frame modulation control register
P_LCD_Palette_Ctrl	0x7989	LCD Palette control register
P_LCD_Attri_Ctrl	0x798A	LCD attribute control register
P_LCD_Palette[0:255]*	0x7A00~0x7AFF	LCD 256-color palette entries

P_LCD_Setup
0x7980
LCD Setup Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	FPIF/C	FPIEN	LCDEN	SELF	PSEL	-	BUSW	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Clock
0x7981
LCD Clock Generation Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Segment
0x7982
LCD Segment Number Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

P_LCD_Common
0x7983
LCD Common Number Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Buffer_LowAdr 0x7984
LCD Buffer Address A15 ~ A0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Buffer_HighAdr 0x7985
LCD Buffer Address A25~A16

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Buffer_Offset 0x7986
LCD Offset Size

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	LCDOFST									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Timing_Ctrl 0x7987
LCD Control Signal Timing Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	LBVL			LPW				LPCPD				
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Frame_Ctrl 0x7988
LCD Frame Modulation Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BCMOD	-	-	-	-	-	-	-	MVAL							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Palette_Ctrl 0x7989
LCD Palette Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	OVIF/C	-	-	-	-	-	-	-	BPR	-	-	-	BPP			LCDBW
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_LCD_Attri_Ctrl 0x798A
LCD Attribute Control Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	VerINV	HORINV	DATAINV	NEGFILE	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TFT LCD Control Register Summary Table

Name	Address	Description
P_TFT_CTRL	0x7D00	TFT Control Register
P_TFT_DCLK_CTRL	0x7D01	TFT Data Clock Control Register
P_TFT_INT_CTRL	0x7D02	TFT Interrupt Control Register
P_TFT_H_WIDTH	0x7D03	TFT Horizontal Width
P_TFT_H_START	0x7D04	TFT Horizontal Start Location
P_TFT_H_END	0x7D05	TFT Horizontal End Location
P_TFT_HSYNC_SETUP	0x7D06	TFT Hsync Setup Register
P_TFT_V_WIDTH	0x7D07	TFT Vertical Width
P_TFT_V_START	0x7D08	TFT Vertical Start Location
P_TFT_V_END	0x7D09	TFT Vertical End Location
P_TFT_VSYNC_SETUP	0x7D0A	TFT Vsync Setup Register
P_TFT_RGB_CTRL	0x7D0B	TFT RGB Mode Control Register
P_TFT_YUV_CTRL	0x7D0C	TFT YUV Mode Control Register
P_TFT_DMASTART_AH	0x7D0D	TFT DMA Start High Address
P_TFT_DMASTART_AL	0x7D0E	TFT DMA Start Low Address

Name	Address	Description
P_TFT_DMA_OFFSET	0x7D0F	TFT DMA Offset Address
P_TFT_PIXEL_NUM	0x7D10	TFT Pixel Numbers in Each Line
P_TFT_LINE_NUM	0x7D11	TFT Line Number in Each Frame
P_TFT_PIP0_CTRL	0x7D12	TFT PIP0 Control Register
P_TFT_PIP0_VIR_SAH	0x7D13	TFT PIP0 Virtual Frame Buffer Start High Address
P_TFT_PIP0_VIR_SAL	0x7D14	TFT PIP0 Virtual Frame Buffer Start Low Address
P_TFT_PIP0_VIR_EAH	0x7D15	TFT PIP0 Virtual Frame Buffer End High Address
P_TFT_PIP0_VIR_EAL	0x7D16	TFT PIP0 Virtual Frame Buffer End Low Address
P_TFT_PIP0_STARTAH	0x7D17	TFT PIP0 Frame Buffer Start High Address
P_TFT_PIP0_STARTAL	0x7D18	TFT PIP0 Frame Buffer Start Low Address
P_TFT_PIP0_H_START	0x7D19	TFT PIP0 Horizontal Start Location in Each Line
P_TFT_PIP0_H_END	0x7D1A	TFT PIP0 Horizontal End Location in Each Line
P_TFT_PIP0_V_START	0x7D1B	TFT PIP0 Vertical Start Location in Each Frame
P_TFT_PIP0_V_END	0x7D1C	TFT PIP0 Vertical End Location in Each Frame
P_TFT_PIP1_CTRL	0x7D1D	TFT PIP1 Control Register
P_TFT_PIP1_VIR_SAH	0x7D1E	TFT PIP1 Virtual Frame Buffer Start High Address
P_TFT_PIP1_VIR_SAL	0x7D1F	TFT PIP1 Virtual Frame Buffer Start Low Address
P_TFT_PIP1_VIR_EAH	0x7D20	TFT PIP1 Virtual Frame Buffer End High Address
P_TFT_PIP1_VIR_EAL	0x7D21	TFT PIP1 Virtual Frame Buffer End Low Address
P_TFT_PIP1_STARTAH	0x7D22	TFT PIP1 Frame Buffer Start High Address
P_TFT_PIP1_STARTAL	0x7D23	TFT PIP1 Frame Buffer Start Low Address
P_TFT_PIP1_H_START	0x7D24	TFT PIP1 Horizontal Start Location in Each Line
P_TFT_PIP1_H_END	0x7D25	TFT PIP1 Horizontal End Location in Each Line
P_TFT_PIP1_V_START	0x7D26	TFT PIP1 Vertical Start Location in Each Frame
P_TFT_PIP1_V_END	0x7D27	TFT PIP1 Vertical End Location in Each Frame
P_TFT_PIP2_CTRL	0x7D28	TFT PIP2 Control Register
P_TFT_PIP2_VIR_SAH	0x7D29	TFT PIP2 Virtual Frame Buffer Start High Address
P_TFT_PIP2_VIR_SAL	0x7D2A	TFT PIP2 Virtual Frame Buffer Start Low Address
P_TFT_PIP2_VIR_EAH	0x7D2B	TFT PIP2 Virtual Frame Buffer End High Address
P_TFT_PIP2_VIR_EAL	0x7D2C	TFT PIP2 Virtual Frame Buffer End Low Address
P_TFT_PIP2_STARTAH	0x7D2D	TFT PIP2 Frame Buffer Start High Address
P_TFT_PIP2_STARTAL	0x7D2E	TFT PIP2 Frame Buffer Start Low Address
P_TFT_PIP2_H_START	0x7D2F	TFT PIP2 Horizontal Start Location in Each Line
P_TFT_PIP2_H_END	0x7D30	TFT PIP2 Horizontal End Location in Each Line
P_TFT_PIP2_V_START	0x7D31	TFT PIP2 Vertical Start Location in Each Frame
P_TFT_PIP2_V_END	0x7D32	TFT PIP2 Vertical End Location in Each Frame
P_TFT_PIP3_CTRL	0x7D33	TFT PIP3 Control Register
P_TFT_PIP3_VIR_SAH	0x7D34	TFT PIP3 Virtual Frame Buffer Start High Address
P_TFT_PIP3_VIR_SAL	0x7D35	TFT PIP3 Virtual Frame Buffer Start Low Address
P_TFT_PIP3_VIR_EAH	0x7D36	TFT PIP3 Virtual Frame Buffer End High Address
P_TFT_PIP3_VIR_EAL	0x7D37	TFT PIP3 Virtual Frame Buffer End Low Address

Name	Address	Description
P_TFT_PIP3_STARTAH	0x7D38	TFT PIP3 Frame Buffer Start High Address
P_TFT_PIP3_STARTAL	0x7D39	TFT PIP3 Frame Buffer Start Low Address
P_TFT_PIP3_H_START	0x7D3A	TFT PIP3 Horizontal Start Location in Each Line
P_TFT_PIP3_H_END	0x7D3B	TFT PIP3 Horizontal End Location in Each Line
P_TFT_PIP3_V_START	0x7D3C	TFT PIP3 Vertical Start Location in Each Frame
P_TFT_PIP3_V_END	0x7D3D	TFT PIP3 Vertical End Location in Each Frame

P_TFT_CTRL					0x7D00		TFT Control Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TFTEN	-	-	-	VS_TYPE	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_DCLK_CTRL								0x7D01		TFT Data Clock Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	DCLK_INV	-	-	-	DCLK_SEL					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	

P_TFT_INT_CTRL		0x7D02				TFT Interrupt Control Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	UF_F/C	UF_EN	-	-	FE_F/C	FE_EN	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_H_WIDTH					0x7D03		TFT Horizontal Width										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	H_WIDTH												
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_H_START		0x7D04								TFT Horizontal Start Location							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	H_START											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_H_END				0x7D05				TFT Horizontal End Location								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	H_END											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_HSYNC_SETUP		0x7D06					TFT Hsync Setup Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HS_POL	-	-	-	-	HS_WIDTH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_V_WIDTH		0x7D07						TFT Vertical Width									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	V_WIDTH									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_V_START		0x7D08							TFT Vertical Start Location								
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	V_START								
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_V_END		0x7D09						TFT Vertical End Location									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	V_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_VSYNC_SETUP								0x7D0A		TFT Vsync Setup Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	VS_POL	-	-	-	-	-	-	VS_WIDTH									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_RGB_CTRL		0x7D0B									TFT RGB Mode Control Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	RGB_M	RGB_DMEN	-	-	-	-	-	-	-	ODD_L_TYPE		-	EVEN_L_TYPE					
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

P_TFT_YUV_CTRL		0x7D0C				TFT YUV Mode Control Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	YUV_EN	YUV_M	CCIR656_EN	-	-	-	-	-	-	-	-	SHARE	-	-	YUV_TYPE		
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	

P_TFT_DMASTART_AH		0x7D0D					TFT DMA Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	DMA_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_DMASTART_AL				0x7D0E				TFT DMA Start Low Address								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DMA_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_DM_OFFSET					0x7D0F		TFT DMA Offset Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	DMA_OFFSET										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

P_TFT_PIXEL_NUM		0x7D10						TFT Pixel Numbers in Each Line								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIXEL_NUM									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_LINE_NUM		0x7D11						TFT Line Numbers in Each Line									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	LINE_NUM									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_CTRL		0x7D12		TFT PIP0 Control Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP0EN	PIP0SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP1_CTRL		0x7D1D				TFT PIP1 Control Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP1EN	PIP1SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_CTRL		0x7D28		TFT PIP2 Control Register													
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		PIP2EN	PIP2SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_CTRL		0x7D33		TFT PIP3 Control Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP3EN	PIP3SCREN	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP0_VIR_SAH				0x7D13			TFT PIP0 Virtual Frame Buffer Start High Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	PIP0_VIR_SAH											
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP1_VIR_SAH					0x7D1E		TFT PIP1 Virtual Frame Buffer Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_SAH					0x7D29		TFT PIP2 Virtual Frame Buffer Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_SAH					0x7D34		TFT PIP3 Virtual Frame Buffer Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP3_VIR_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_VIR_SAL				0x7D14				TFT PIP0 Virtual Frame Buffer Start Low Address											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	PIP0_VIR_SAL																		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_TFT_PIP1_VIR_SAL				0x7D1F			TFT_PIP1 Virtual Frame Buffer Start Low Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_SAL						0x7D2A		TFT_PIP2_Virtual_Frame_Buffer_Start_Low_Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP2_VIR_SAL																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP3_VIR_SAL				0x7D35			TFT_PIP3_Virtual_Frame_Buffer_Start_Low_Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP3_VIR_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_VIR_EAH					0x7D15		TFT PIP0 Virtual Frame Buffer End High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP0_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_EAH					0x7D20		TFT PIP1 Virtual Frame Buffer End High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_VIR_EAH					0x7D2B		TFT PIP2 Virtual Frame Buffer End High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_VIR_EAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_VIR_EAL				0x7D16				TFT PIP0 Virtual Frame Buffer End Low Address								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_VIR_EAL				0x7D21				TFT PIP1 Virtual Frame Buffer End Low Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP1_VIR_EAL																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_VIR_EAL				0x7D2C			TFT_PIP2_Virtual_Frame_Buffer_End_Low_Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP2_VIR_EAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_VIR_EAL						0x7D37		TFT_PIP3_Virtual_Frame_Buffer_End_Low_Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	PIP3_VIR_EAL																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP0_STARTAH					0x7D17		TFT PIP0 Frame Buffer Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP0_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_STARTAH					0x7D22		TFT PIP1 Frame Buffer Start High Address									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP1_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_STARTAH 0x7D2D						TFT PIP2 Frame Buffer Start High Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP2_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_STARTAH 0x7D38						TFT PIP3 Frame Buffer Start High Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	PIP3_SAH										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_STARTAL 0x7D18						TFT PIP0 Frame Buffer Start Low Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP0_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_STARTAL 0x7D23						TFT PIP1 Frame Buffer Start Low Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP1_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_STARTAL 0x7D2E						TFT PIP2 Frame Buffer Start Low Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP2_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_STARTAL 0x7D39						TFT PIP3 Frame Buffer Start Low Address										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	PIP3_SAL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_H_START 0x7D19						TFT PIP0 Horizontal Start Location in Each Line										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_H_START 0x7D24						TFT PIP1 Horizontal Start Location in Each Line										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP1_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_H_START			0x7D2F				TFT PIP2 Horizontal Start Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP2_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_H_START			0x7D3A				TFT PIP3 Horizontal Start Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP3_H_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_H_END				0x7D1A			TFT PIP0 Horizontal End Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_H_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_H_END				0x7D25			TFT PIP1 Horizontal End Location in Each Line										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP1_H_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_H_END				0x7D30				TFT PIP2 Horizontal End Location in Each Line									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP2_H_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP3_H_END			0x7D3B				TFT PIP3 Horizontal End Location in Each Line										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP3_H_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP0_V_START			0x7D1B				TFT PIP0 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_V_START				0x7D26			TFT PIP1 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP1_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP2_V_START			0x7D31				TFT PIP2 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP2_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_V_START				0x7D3C			TFT PIP3 Vertical Start Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP3_V_STR									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP0_V_END				0x7D1C			TFT PIP0 Vertical End Location in Each Frame									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	PIP0 V_END									
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP1_V_END				0x7D27			TFT PIP1 Vertical End Location in Each Frame										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	PIP1_V_END										
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_TFT_PIP2_V_END		0x7D32						TFT PIP2 Vertical End Location in Each Frame									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	PIP2_V_END									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TFT_PIP3_V_END		0x7D3D						TFT PIP3 Vertical End Location in Each Frame									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	PIP3_V_END									
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UART/IrDA Control Register Summary Table

Name	Address	Description
P_UARTIrDA_Data	0x7900	UART/ IrDA Data Register
P_UART_RXStatus	0x7901	UART Reception Status (Error Flag) Register
P_UARTIrDA_Ctrl	0x7902	UART/ IrDA Control Register
P_UART_BaudRate	0x7903	UART Baud Rate Setup Register
P_UARTIrDA_Status	0x7904	UART/ IrDA Interrupt Register
P_UARTIrDA_FIFO	0x7905	UART/IrDA FIFO Control Register
P_UART_TXDLY	0x7906	UART TX Delay Control Register
P_IrDA_BaudRate	0x7907	IrDA Baud Rate Setup Register
P_IrDA_Ctrl	0x7908	IrDA Control Register
P_IrDALP	0x7909	IrDA Low Power Control Register

P_UARTIrDA_Data		0x7900								UART / IrDA Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	UARTDATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_UART_RXStatus				0x7901				UART Reception Error Flag Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	-	-	-	-	-	OE	BE	PE	FE	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_UARTIrDA_Ctrl			0x7902					UART / IrDA Control Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	RXIE	TXIE	RTIE	UEN	MSIE	SLT	-	-	-	WLSEL	FEN	SBSEL	PSEL	PEN	SB		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_UART_BaudRate	0x7903										UART Baud Rate Setup Register						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	BUAD																
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_UARTIrDA_Status		0x7904				UART / IrDA Status Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		RXIF	TXIF	RTIF	-	-	-	-	-	TXEF	RXFF	TXFF	RXEF	BY	DCD	DSR	CTS
Default		0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0

P_UARTIrDA_FIFO		0x7905				UART/IrDA FIFO Control Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	TX_LEVEL			-	TX_FLAG			-	RX_LEVEL			-	RX_FLAG		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_UART_TXDLY		0x7906								UART TX Delay Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	TWT			
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IrDA_BaudRate		0x7907								IrDA Baud Rate Setup Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		BUAD															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_IrDA_Ctrl		0x7908								IRDA Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TXLT				TPOL	RPOL	IEN	ILP	RXLT							
Default		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

P_IrDALP		0x7909								IrDA Low Power Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	IrDALP							
Default		0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

SPI Control Register Summary Table

Name	Address	Description
P_SPI_Ctrl	0x7940	SPI Control Register
P_SPI_TXStatus	0x7941	SPI Transmit Status Register
P_SPI_TXData	0x7942	SPI Transmit FIFO Register
P_SPI_RXStatus	0x7943	SPI Receive Status Register
P_SPI_RXData	0x7944	SPI Receive FIFO Register
P_SPI_Misc	0x7945	SPI Misc Control Register

P_SPI_Ctrl		0x7940								SPI Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SPIEN	-	LBM	-	SPIRST	-	-	MOD	-	-	SCKPHA	SCKPOL	-	SCKSEL		
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SPI_TXStatus		0x7941								SPI Transmit Status Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SPITXIF	SPITXIEN	-	-	-	-	-	-	TXFLEV				TXFFLAG			
Default		0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

P_SPI_TXData		0x7942								SPI Transmit FIFO Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	SPIDATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SPI_RXStatus		0x7943							SPI Transmit Status Register								
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SPITXIF	SPITXIEN	-	-	-	-	RXFULL	RXFOV	TXFLEV			TXFFLAG				
Default		0	0							0	0	0	0	0	0	0	0

P_SPI_RXData		0x7944								SPI Receive FIFO Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	SPIDATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SPI_Misc		0x7945						SPI Misc. Control Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	OVER	SMART	-	-	-	BSY	RFF	RNE	TNF	TFE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

USB Device Register Summary Table

Name	Address	Description
P_USBD_Config	0x7B30	USB Configuration Register
P_USBD_Device	0x7B57	USB Device Register
P_USBD_Function	0x7B31	USB Function Register
P_USBD_DMAINT	0x7B59	USB DMA Interrupt Register
P_USBD_PMR	0x7B32	USB Power Management Register
P_USBD_EP0Data	0x7B33	USB Endpoint0 Data Register
P_USBD_BIData	0x7B34	USB Bulk In Data Register
P_USBD_BOData	0x7B35	USB Bulk Out Data Register
P_USBD_INTINData	0x7B36	USB Interrupt In Data Register
P_USBD_NullPkt	0x7B58	USB Null Packet Register
P_USBD_EPEvent	0x7B37	USB Endpoint Event Register
P_USBD_GLOINT	0x7B38	USB Global Interrupt Register
P_USBD_INTEN	0x7B39	USB Interrupt Enable Register
P_USBD_INTF	0x7B3A	USB Interrupt Flag Register
P_USBD_SCI NTEN	0x7B3B	USB Standard Command Interrupt Enable Register
P_USBD_SCINTF	0x7B3C	USB Standard Command Interrupt Flag Register
P_USBD_EPAutoSet	0x7B3D	USB Endpoint Auto Set Register
P_USBD_EPSetStall	0x7B3E	USB Endpoint Set Stall Register
P_USBD_EPBufClear	0x7B3F	USB Endpoint Buffer Clear Register
P_USBD_EPEvntClear	0x7B40	USB Endpoint Event Clear Register
P_USBD_EP0WrtCount	0x7B41	USB Endpoint0 Write Count Register
P_USBD_BOWrtCount	0x7B42	USB Bulk Out Write Count Register
P_USBD_EP0BufPointer	0x7B43	USB Endpoint0 Buffer Pointer Register
P_USBD_BIBufPointer	0x7B44	USB Bulk In Buffer Pointer Register
P_USBD_BOBufPointer	0x7B45	USB Bulk Out Buffer Pointer Register
P_USBD_EP0RTR	0x7B46	USB Endpoint0 bmRequestType Register
P_USBD_EP0RR	0x7B47	USB Endpoint0 bRequest Register

Name	Address	Description
P_USBD_EP0VR	0x7B48	USB Endpoint0 wValue Register
P_USBD_EP0IR	0x7B49	USB Endpoint0 wIndex Register
P_USBD_EP0LR	0x7B4A	USB Endpoint0 wLength Register
P_USBD_DMAWrtCountL	0x7B50	USB DMA Byte Count Low Register
P_USBD_DMAWrtCountH	0x7B51	USB DMA Byte Count High Register
P_USBD_DMAAckL	0x7B52	USB DMA ACK Count Low Register
P_USBD_DMAAckH	0x7B53	USB DMA ACK Count High Register
P_USBD_EPStall	0x7B54	USB Endpoint Stall Bit Register

USB Host Register Summary Table

Name	Address	Description
P_USBH_Config	0x7B00	USB Host Configuration Register
P_USBH_TimeConfig	0x7B01	USB Host Timing Configuration Register
P_USBH_Data	0x7B02	USB Host Data Register
P_USBH_Transfer	0x7B03	USB Host Transfer Register
P_USBH_DveAddr	0x7B04	USB Device Address Register
P_USBH_DveEP	0x7B05	USB Device Endpoint Register
P_USBH_TXCount	0x7B06	USB Host Transmit Count Register
P_USBH_RXCount	0x7B07	USB Receive Count Register
P_USBH_FIFOInPointer	0x7B08	USB Host FIFO Input Pointer Register
P_USBH_FIFOOOutPointer	0x7B09	USB Host FIFO Output Pointer Register
P_USBH_AutoInByteCount	0x7B0A	USB Host Automatic In Transaction Byte Count Register
P_USBH_AutoOutByteCount	0x7B0B	USB Host Automatic Out Transaction Byte Count Register
P_USBH_AutoTrans	0x7B0C	USB Host Auto Transfer Register
P_USBH_Status	0x7B0D	USB Host Status Register
P_USBH_INTF	0x7B0E	USB Host Interrupt Flag Register
P_USBH_INTEN	0x7B0F	USB Host Interrupt Enable Register
P_USBH_StorageRST	0x7B10	USB Storage Reset Register
P_USBH_SoftRST	0x7B11	USB Software Reset Register / Device Plug Out Register
P_USBH_SOFTimer	0x7B12	USB SOF Timer Register
P_USBH_FrameNum	0x7B13	USB Frame Number Register
P_USBH_OTGConfig	0x7B14	USB OTG Configuration Register
P_USBH_VBusSet	0x7B15	USB VBUS Set Register
P_USBH_VbusStatus	0x7B16	USB VBUS Status Register
P_USBH_INAckCount	0x7B17	USB IN ACK Count Register
P_USBH_OutAckCount	0x7B18	USB OUT ACK Count Register
P_USBH_RSTAckCount	0x7B19	USB Reset ACK Count Register

Name	Address	Description
P_USBH_Storage1/2	0x7B1A	For Debugging
P_USBH_DReadback	0x7B1B	USB D+ / D- Readback Register

P_USBD_Config 0x7B30 USB Configuration Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	RWUPEN	SPWR	USBEN	TNSPL	TNSPH	BYPASS
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_Device 0x7B57 USB Device Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP4_Type	EP3_Type	EP2_Type	EP1_Type	EP4_IO	EP3_IO	EP2_IO	EP1_IO	-	-	-	MOD	-	-	-	-
Default	0	1	1	1	1	0	1	0	0	1	0	1	0	0	0	0

P_USBD_Function 0x7B31 USB Function Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	SRST	DMA_BOEN	DMA_BIEN	Config_Value	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_DMAINT 0x7B59 USB DMA Interrupt Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	DMAINTEN_CLR	DMAINTEN	DMAINTF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_PMR 0x7B32 USB Power Management Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	RESWKE	RE_WA	RE_WAFE	RST	SUS_Mod
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EP0Data 0x7B33 USB Endpoint0 Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_BIData 0x7B34 USB Bulk IN Data Register

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_BODData		0x7B35								USB Bulk OUT Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	BODATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_INTINData		0x7B36								USB Interrupt IN Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	INTINDATA							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_NullPkt		0x7358												USB Null Packet Register				
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2		1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	IIN_NULLPKT		BI_NULLPKT	EP0_NULLPKT
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0

P_USBD_EPEvent		0x7B37								USB Endpoint Event Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		IINNA	IINPR	BONA	BOPR	BOPE	BINA	BIPC	BIPR	E0SNA	E0SEN	E0INNA	E0INPR	E0ONA	E0OPR	E0OPE	E0SPR
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_GLOINT		0x7B38								USB Global Interrupt Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	DMA	STANDARD	POWER	INT	BO	BI	EP0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_INTEN		0x7B39								USB Interrupt Enable Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_INTF		0x7B3A								USB Interrupt Flag Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		RST	RME	SUS	IINNA	IINPC	BONA	BOPS	BINA	BIPC	E0SNA	E0SC	E0INNA	E0INPC	E0ONA	E0OPS	E0SPS
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_SCINTEN		0x7B3B								USB Standard Command Interrupt Enable Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_SCINTF	0x7B3C								USB Standard Command Interrupt Flag Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	GSTS	CFEA	SFEA	SADD	GCON	SCON	GINT	SINT
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EPAutoSet	0x7B3D								USB Endpoint Auto Set Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	EP0ASE	IAINPR	BAOPE	BAIPR	E0AIPR	E0AOPE
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EPSetStall	0x7B3E								USB Endpoint Set Stall Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	IINS	BOS	BIS	EP0S
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EPBufClear	0x7B3F								USB Endpoint Buffer Clear Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	IINBC	BOBC	BIBC	EP0BC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EPEvntClear	0x7B40								USB Endpoint Event Clear Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	IINPC	BOEC	BIPC	EP0SC	EP0IPC	EP0OEC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EP0WrtCount	0x7B41								USB Endpoint0 Write Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	EP0WC		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_BOWrtCount	0x7B42								USB Bulk OUT Write Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	BOWC		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD_EP0BufPointer	0x7B43								USB Endpoint0 Buffer Pointer Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	EP0WBP			EP0RBP		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	BIBufPointer	0x7B44	USB Bulk IN Buffer Pointer Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BIBWP								BIBRP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	BOBufPointer	0x7B45	USB Bulk OUT Buffer Pointer Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	BOBWP								BOBRP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	EP0RTR	0x7B46	USB Endpoint0 bmRequestType Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	EP0RTR							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	EP0RR	0x7B47	USB Endpoint0 bRequest Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	EP0RR							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	EP0VR	0x7B48	USB Endpoint0 wValue Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP0VR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	EP0IR	0x7B49	USB wIndex Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP0IR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	EP0LR	0x7B4A	USB Endpoint0 wLength Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	EP0LR															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBD	DMAWrtCountL	0x7B50	USB DMA Byte Count Low Register													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DMAWCL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB	DMAWrtCountH	0x7B51							USB DMA Byte Count High Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	DMAWCH							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB_DMAACK	0x7B52							USB DMA ACK Count Low Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	DMAACKL															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB	DMAACK 0x7B53 USB DMA ACK Count High Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	DMAACKH		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB	EPStall				0x7B54				USB Endpoint Stall Bit Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	IISS	BOSS	BISS	EPOSS	-	-	-	-	IISB	BOSB	BISB	EP0SB
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB_Config	0x7B00 USB Host Configuration Register															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	SUS	-	ASOF	SOFR	-	HOSTEN
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB_TimeConfig				0x7B01 USB Host Timing Configuration Register												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	SAU	PAC	TC		IPD	
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USB	Data 0x7B02								USB Host Data Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	HDATA							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH Transfer			0x7B03				USB Host Transfer Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	RST	OD1	OD0	ID1	ID0	Setup	SOF
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_DveAddr	0x7B04								USB Device Address Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	DAddr						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_DveEP	0x7B05								USB Device Endpoint Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	DEP			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_TXCount	0x7B06								USB Host Transmit Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	TXC						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_RXCount	0x7B07								USB Host Receive Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	RXC						
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_FIFOInPointer	0x7B08								USB Host FIFO Input Pointer Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	HFIP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_FIFOPutPointer	0x7B09								USB Host FIFO Output Pointer Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	HFOP							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_AutoInByteCount	0x7B0A								USB Host Automatic In Transaction Byte Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HAIBC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_AutoOutByteCount	0x7B0B								USB Host Automatic Out Transaction Byte Count Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	HAOBC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_AutoTrans			0x7B0C				USB Host Auto Transfer Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	CAO	CAI	AOX	AIX
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_Status			0x7B0D				USB Host Status Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	TO	CRC	DE	BS	UP	SH	NH	AH
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_INTF			0x7B0E				USB Host Interrupt Flag Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_INTEN			0x7B0F				USB Host Interrupt Enable Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	DPO	TRST	TSOFI	ITOK	TXO	VSC	AOX	AIX	RX	TX	SOF	DSC
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_SoftRST			0x7B11				USB Software Reset Register / Device Plug Out Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	DPOE	DPOTV							SRST
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_INAckCount			0x7B17				USB IN ACK Count Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	INACK															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_OutAckCount			0x7B18				USB OUT ACK Count Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	OUTACK															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_RSTAckCount			0x7B19				USB Reset ACK Count Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	IARST	OARST
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_USBH_Dreadback					0x7B1B												USB D+ / D- Readback Register	
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	-	DM	DP		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

ADC control register summary table

Name	Address	Description
P_ADC_Setup	0x7960	ADC Setup Register
P_MADC_Ctrl	0x7961	Manual Mode ADC Control Register
P_MADC_Data	0x7962	Manual Mode ADC Data Register
P_ASADC_Ctrl	0x7963	Auto Sample Mode ADC Control Register
P_ASADC_Data	0x7964	Auto Sample Mode ADC Data Register
P_TP_Ctrl	0x7965	Touch Panel Control Register
P_HQADC_Ctrl	0x7970	High Quality ADC control
P_HQADC_PGAS	0x7971	High Quality ADC MICIN pre gain setting
P_HQADC_RGAIN	0x7972	High Quality ADC LINEINR gain setting
P_HQADC_LGAIN	0x7973	High Quality ADC LINEINL gain setting

P_ADC_Setup		0x7960					ADC Setup Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ADBEN	ADCEN	-	-	-	CLKSEL		ASEN		-	-	-	-	ASMEN		
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_MADC_Ctrl				0x7961 Manual Mode ADC Control Register												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ADCRIF/C	ADCRIEN	-	-	-	-	-	-	CNVRDY	STRCNV	-	-	-	CHSEL		
Default	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

P_MADC Data				0x7962				Manual ADC Data Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	MADCDA												-	-	-	-			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_ASADC_Ctrl						0x7963											Auto Sample control register															
Bit	15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
Function	ASIF/C		ASIEN		ASFF		ASFOV		DMA		OVER		ASFIL								FIFOLEV											
Default	0		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0	

P_ASADC_Data				0x7964				Auto Sample Data register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	ASADC															
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TP_Ctrl		0x7965						Touch Panel Control Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		TPIF/C	TPIEN	TPEN	TPST	TMOD	-	-	-	-	-	-	DBEN	-	-	DBTSEL	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_HQADC_Ctrl		0x7970								High Quality ADC control							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	DIV_REC			MONO	BOOST	INMODE		PWADL	PWADR	MICBIAS	ADMCLK
Default		0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

P_HQADC_PGAS			0x7971						High Quality ADC MICIN pre gain setting								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	-	-	-	-	-	PGAS					
Default	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	

P_HQADC_RGAIN		0x7972				High Quality ADC LINEINR gain setting											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	ADROV	ADROV_IEN	-	-	-	-	-	-	-	-	-	LINEGR					
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	

P_HQADC_LGAIN		0x7973				High Quality ADC LINEINL gain setting											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	ADLOV	ADLOV_IEN	-	-	-	-	-	-	-	-	-	LINEGL					
Default	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	

Nand Flash Control Register Summary Table

Name	Address	Description
P_NF_Ctrl	0x7850	NAND Flash Control Register
P_NF_CMD	0x7851	NAND Flash Command Register
P_NF_AddrL	0x7852	NAND Flash Low Address Register
P_NF_AddrH	0x7853	NAND Flash High Address Register
P_NF_Data	0x7854	NAND Flash Data Register
P_NF_INT_Ctrl	0x7855	NAND Flash DMA / INT Control Register
P_ECC_Ctrl	0x7857	ECC Control Register
P_ECC_LPRL_LB	0x7858	ECC Low Byte Line Parity LSB Register
P_ECC_LPRH_LB	0x7859	ECC Low Byte Line Parity MSB Register
P_ECC_CPR_LB	0x785A	ECC Low Byte Column Parity Check LSB Register
P_ECC_LPR_CKL_LB	0x785B	ECC Low Byte Line Parity Check LSB Register
P_ECC_LPR_CKH_LB	0x785C	ECC Low Byte Line Parity Check MSB Register
P_ECC_CPCKR_LB	0x785D	ECC Low Byte Column Parity Check Register
P_ECC_ERR0_LB	0x785E	ECC Low Byte Error Flag0

Name	Address	Description
P_ECC_ERR1_LB	0x785F	ECC Low Byte Error Flag1
P_ECC_LPRL_HB	0x7848	ECC High Byte Line Parity LSB Register
P_ECC_LPRH_HB	0x7849	ECC High Byte Line Parity MSB Register
P_ECC_CPR_HB	0x784A	ECC High Byte Column Parity Register
P_ECC_LPR_CKL_HB	0x784B	ECC High Byte Line Parity Check LSB Register
P_ECC_LPR_CKH_HB	0x784C	ECC High Byte Line Parity Check MSB Register
P_ECC_CPCKR_HB	0x784D	ECC High Byte Column Parity Check Register
P_ECC_ERR0_HB	0x784E	ECC High Byte Error Flag0
P_ECC_ERR1_HB	0x784F	ECC High Byte Error Flag1
P_CHECKSUM0_LB	0x7830	NAND Flash Low Byte Check Sum Low Value
P_CHECKSUM1_LB	0x7831	NAND Flash Low Byte Check Sum High Value
P_CHECKSUM0_HB	0x7832	NAND Flash High Byte Check Sum Low Value
P_CHECKSUM1_HB	0x7833	NAND Flash High Byte Check Sum High Value

P_NF_Ctrl	0x7850	NAND Flash Control Register
Bit	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
Function	NFBF 8or16 - - - - - - NFC7 NFC6 NFC5 NFC4 NFC3 NFC2 NFC1 NFC0	
Init	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	

P_NF_CMD		0x7851								NAND Flash Command Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	NFCMD																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_NF_AddrL	0x7852								NAND Flash ADDR Low Word Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	2 nd Cycle								1 st Cycle							
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_NF_AddrH	0x7853								NAND Flash ADDR High Word Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	4 th Cycle								3 rd Cycle							
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_NF_Data				0x7854				NAND Flash Data Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	NFDATA															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_NF_INT_Ctrl				0x7855				DMA/INT Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	REQF/C	DMAEN	INTEN	-	ADR4EN	ADR3EN	ADR2EN	-	-	-	-	-	-	-	-	-
Init	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0

P_ECC_Ctrl				0x7857				ECC Control Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	-	-	-	-	-	ECCSPT	CKP	ERST
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_LPRL_LB				0x7858				ECC Low Byte Line parity LSB Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_LPRH_LB				0x7859				ECC Low Byte Line parity MSB Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_CPR_LB			0x785A						ECC Low Byte Column parity Register							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	CPR											
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_LPR_CKL_LB				0x785B				ECC Low Byte Line parity Check LSB Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
Function	LPRCKL																		
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

P_ECC_LPR_CKH_LB				0x785C				ECC Low Byte Line parity Check MSB Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRCKH															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_CPCKR_LB			0x785D				ECC Low Byte Column parity Check Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	CPRCK											
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_ERR0_LB				0x785E				ECC Low Byte Field-0 Error Flag									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE								
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	

P_ECC_ERR1_LB			0x785F					ECC Low Byte Field-1 Error Flag									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE								
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	

P_ECC_LPRL_HB		0x7848				ECC High Byte Line parity LSB Register										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	LPRL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_ECC_LPRH_HB		0x7849				ECC High Byte Line parity MSB Register											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	LPRH																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_ECC_CPR_HB			0x784A						ECC High Byte Column parity Register								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	CPR												
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_ECC_LPR_CKL_HB					ECC High Byte Line parity Check LSB Register												
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	LPRCKL																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_ECC_LPR_CKH_HB				0x784C				ECC High Byte Line parity Check MSB Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	LPRCKH																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_ECC_CPCR_HB			0x784D					ECC High Byte Column parity Check Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	CPRCK												
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_ECC_ERR0_HB					0x784E		ECC High Byte Field-0 Error Flag										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE								
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	

P_ECC_ERR1_HB						0x784F											ECC High Byte Field-1 Error Flag										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0											
Function	-	-	-	2ERR	1ERR	FAILBIT			FAILLINE																		
Init	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1											

P_CHECKSUM0_LB					NAND Flash Low Byte Check Sum Low Value											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM0_LB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_CHECKSUM1_LB					0x7831												NAND Flash Low Byte Check Sum High Value											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0												
Function	CHECKSUM1_LB																											
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0												

P_CHECKSUM0_HB					NAND Flash High Byte Check Sum Low Value											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM0_HB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_CHECKSUM1_HB					NAND Flash High Byte Check Sum High Value											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	CHECKSUM1_HB															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2C Register Summary Table

Name	Address	Description
P_I2C_Ctrl	0x7B60	I2C Control Register
P_I2C_Status	0x7B61	I2C Status Register
P_I2C_Addr	0x7B62	I2C Address Register
P_I2C_Data	0x7B63	I2C Data Register
P_I2C_DeCLK	0x7B64	I2C De-Bounce Clock Register
P_I2C_En	0x7B65	I2C Interface Enable Register

P_I2C_Ctrl		0x7B60								I2C Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	ACKEN	CLKSEL	INTEN	INTPEND/C	TXCLK			
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_I2C_Status		0x7B61								I2C Status Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Mod		BY	DataEN	ArbS	SS	AddrS	LS
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_I2C_Addr		0x7B62								I2C Address Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Addr							-
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_I2C_Data		0x7B63								I2C Data Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data							
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_I2C_DeCLK		0x7B64								I2C De-bounce Clock Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	DEBCLK							
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_I2C_En		0x7B65								I2C Enable Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	I2CEN
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA Control Register Summary Table

Name	Address	Description
P_DMA_Ctrl0	0x7B80	DMA Channel Control Register 0
P_DMA_SRC_AddrL0	0x7B81	DMA Source Low Address [15:0] Register 0
P_DMA_TAR_AddrL0	0x7B82	DMA Target Low Address [15:0] Register 0
P_DMA_TCountL0	0x7B83	DMA Terminal Counter Low [15:0] Register 0
P_DMA_SRC_AddrH0	0x7B84	DMA Source High Address [25:16] Register 0
P_DMA_TAR_AddrH0	0x7B85	DMA Target High Address [25:16] Register 0
P_DMA_TCountH0	0x7B86	DMA Terminal Counter High [25:16] Register 0
P_DMA_MISC0	0x7B87	DMA miscellaneous Control Register 0
P_DMA_Ctrl1	0x7B88	DMA Channel Control Register 1
P_DMA_SRC_AddrL1	0x7B89	DMA Source Low Address [15:0] Register 1

Name	Address	Description
P_DMA_TAR_AddrL1	0x7B8A	DMA Target Low Address [15:0] Register 1
P_DMA_TCountL1	0x7B8B	DMA Terminal Counter Low [15:0] Register 1
P_DMA_SRC_AddrH1	0x7B8C	DMA Source High Address [25:16] Register 1
P_DMA_TAR_AddrH1	0x7B8D	DMA Target High Address [25:16] Register 1
P_DMA_TCountH1	0x7B8E	DMA Terminal Counter High [25:16] Register 1
P_DMA_MISC1	0x7B8F	DMA miscellaneous Control Register 1
P_DMA_Ctrl2	0x7B90	DMA Channel Control Register 2
P_DMA_SRC_AddrL2	0x7B91	DMA Source Low Address [15:0] Register 2
P_DMA_TAR_AddrL2	0x7B92	DMA Target Low Address [15:0] Register 2
P_DMA_TCountL2	0x7B93	DMA Terminal Counter Low [15:0] Register 2
P_DMA_SRC_AddrH2	0x7B94	DMA Source High Address [25:16] Register 2
P_DMA_TAR_AddrH2	0x7B95	DMA Target High Address [25:16] Register 2
P_DMA_TCountH2	0x7B96	DMA Terminal Counter High [25:16] Register 2
P_DMA_MISC2	0x7B97	DMA miscellaneous Control Register 2
P_DMA_Ctrl3	0x7B98	DMA Channel Control Register 3
P_DMA_SRC_AddrL3	0x7B99	DMA Source Low Address [15:0] Register 3
P_DMA_TAR_AddrL3	0x7B9A	DMA Target Low Address [15:0] Register 3
P_DMA_TCountL3	0x7B9B	DMA Terminal Counter Low [15:0] Register 3
P_DMA_SRC_AddrH3	0x7B9C	DMA Source High Address [25:16] Register 3
P_DMA_TAR_AddrH3	0x7B9D	DMA Target High Address [25:16] Register 3
P_DMA_TCountH3	0x7B9E	DMA Terminal Counter High [25:16] Register 3
P_DMA_MISC3	0x7B9F	DMA miscellaneous Control Register 3
P_DMA_SPRISIZE0	0x7BB0	DMA Sprite Size [9:0] Register 0
P_DMA_SPRISIZE1	0x7BB1	DMA Sprite Size [9:0] Register 1
P_DMA_SPRISIZE2	0x7BB2	DMA Sprite Size [9:0] Register 2
P_DMA_SPRISIZE3	0x7BB3	DMA Sprite Size [9:0] Register 3
P_DMA_TRANSPAT0	0x7BB8	DMA Transparent Pattern Register 0
P_DMA_TRANSPAT1	0x7BB9	DMA Transparent Pattern Register 1
P_DMA_TRANSPAT2	0x7BBA	DMA Transparent Pattern Register 2
P_DMA_TRANSPAT3	0x7BBB	DMA Transparent Pattern Register 3
P_DMA_LINELENGTH	0x7BBD	DMA Line Length Control Register
P_DMA_SS	0x7BBE	DMA Source Select Register
P_DMA_INT	0x7BBF	DMA Interrupt Status Register

P_DMA_Ctrl0		0x7B80						DMA Channel Control Register 0									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_Ctrl1		0x7B88						DMA Channel Control Register 1									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_Ctrl2		0x7B90						DMA Channel Control Register 2									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_Ctrl3		0x7B98						DMA Channel Control Register 3									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		WriteReq	TM	TARByte	SRCByte	TD		RS	CIE	SF	DF	SD	DD	DB/NOR	Mod	BS	CE
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrL0		0x7B81				DMA Source Low Address Register 0											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SRC_Addr															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrL1		0x7B89				DMA Source Low Address Register 1											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	SRC_Addr																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_SRC_AddrL2		0x7B91				DMA Source Low Address Register 2											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	SRC_Addr																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_SRC_AddrL3		0x7B99				DMA Source Low Address Register 3											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		SRC_Addr															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrL0				0x7B82				DMA Target Low Address Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	TAR_Addr																
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TAR_AddrL1	0x7B8A					DMA Target Low Address Register 1										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TAR_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrL2	0x7B92					DMA Target Low Address Register 2										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TAR_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrL3	0x7B9A					DMA Target Low Address Register 3										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TAR_Addr															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL0	0x7B83					DMA Terminal Count Low Register 0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TCountL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL1	0x7B8B					DMA Terminal Count Low Register 1										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TCountL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL2	0x7B93					DMA Terminal Count Low Register 2										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TCountL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountL3	0x7B9B					DMA Terminal Count Low Register 3										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	TCountL															
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH0	0x7B84					DMA Source High Address Register 0										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH1	0x7B8C						DMA Source High Address Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH2	0x7B94						DMA Source High Address Register 2									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SRC_AddrH3	0x7B9C						DMA Source High Address Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	SRC_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrH0	0x7B85						DMA Target High Address Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TAR_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrH1	0x7B8D						DMA Target High Address Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TAR_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrH2	0x7B95						DMA Target High Address Register 2									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TAR_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TAR_AddrH3	0x7B9D						DMA Target High Address Register 3									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TAR_AddrH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountH0	0x7B86						DMA Terminal Count High Register 0									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TCountH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountH1		0x7B8E						DMA Terminal Count High Register 1									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	-	-	-	-	-	-	TCountH										
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_DMA_TCountH2		0x7B96						DMA Terminal Count High Register 2								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TCountH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TCountH3		0x7B9E						DMA Terminal Count High Register 3								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	TCountH									
Init	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_MISC0		0x7B87						DMA miscellaneous Control Register 0									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE		
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_MISC1		0x7B8F						DMA miscellaneous Control Register 1									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE		
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_MISC2		0x7B97						DMA miscellaneous Control Register 2									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE		
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_MISC3		0x7B9F						DMA miscellaneous Control Register 3									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DMARQ	ERRW	-	TRANS_EN	DMATO								-	STATE		
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE0		0x7BB0						DMA Sprite Size [9:0] Register 0									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	SPRISIZE									
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE1 0x7BB1 DMA Sprite Size [9:0] Register 1		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		-	-	-	-	-	-	SPRISIZE									
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE2 0x7BB2 DMA Sprite Size [9:0] Register 2		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		-	-	-	-	-	-	SPRISIZE									
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SPRISIZE3 0x7BB3 DMA Sprite Size [9:0] Register 3		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		-	-	-	-	-	-	SPRISIZE									
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TRANSPAT0 0x7BB8 DMA Transparent Pattern Register 0		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		TRANSPAT															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TRANSPAT1 0x7BB9 DMA Transparent Pattern Register 1		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		TRANSPAT															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TRANSPAT2 0x7BBA DMA Transparent Pattern Register 2		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		TRANSPAT															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_TRANSPAT3 0x7BBB DMA Transparent Pattern Register 3		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		TRANSPAT															
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_LINELENGTH 0x7BBD DMA Line Length Control Register		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit																	
Function		-	-	-	-	-	-	LINELENGTH									
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_DMA_SS		0x7BBE								DMA Source Select Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DMA_SS3				DMA_SS2				DMA_SS1				DMA_SS0			
Init		0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

P_DMA_INT		0x7BBF				DMA Interrupt Register											
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	CH3BY	CH2BY	CH1BY	CH0BY	CH3TOIF	CH2TOIF	CH1TOIF	CH0TOIF	CH3IF	CH2IF	CH1IF	CH0IF
Init		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SD/MMC Register Summary Table

Name	Address	Description
P_SD_DataTX	0x79D0	SD/MMC Data Transmit Register
P_SD_DataRX	0x79D1	SD/MMC Data Receive Register
P_SD_CMD	0x79D2	SD/MMC Command Register
P_SD_ArgL	0x79D3	SD/MMC Argument Low Word Register
P_SD_ArgH	0x79D4	SD/MMC Argument High Word Register
P_SD_RespL	0x79D5	SD/MMC Response Low Word Register
P_SD_RespH	0x79D6	SD/MMC Response High Word Register
P_SD_Status	0x79D7	SD/MMC Status Register
P_SD_Ctrl	0x79D8	SD/MMC Control Register
P_SD_BLKLEN	0x79D9	SD/MMC Block Length Register
P_SD_INT	0x79DA	SD/MMC Interrupt Enable Register

P_SD_DataTX		0x79D0						SD/MMC Data Transmit Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DataTX															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_DataRX		0x79D1								SD/MMC Data Receive Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		DataRX															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_CMD		0x79D2						SD/MMC Command Register									
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	RespType		IniCard	MulBlk	TranData	CmdWD	RunCmd	StpCmd	CmdCode						
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_ArgL		0x79D3								SD/MMC Argument Low Word Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		ArguMentL															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_ArgH		0x79D4								SD/MMC Argument High Word Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		ArguMentH															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_RespL		0x79D5								SD/MMC Response Low Word Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		RespL															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_RespH		0x79D6								SD/MMC Response High Word Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		RespH															
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_Status		0x79D7								SD/MMC Status Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	CINT	CPRE/C	CWP	DCRCE/C	TO/C	DBufEpt	DBufFu	RBufFu	RCRCE/C	RidxE/C	DCOM/C	CCOM/C	CBY	BY
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_Ctrl		0x79D8								SD/MMC Control Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	SDEN	IOEN	DMAMOD	BUSWD	CLKDIV							
Default		0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

P_SD_BLKLEN		0x79D9								SD/MMC Block Length Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	BLKLEN											
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_SD_INT		0x79DA								SD/MMC Interrupt Enable Register							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	IOINT	INSINT	DBULEPT	DBULFU	CBULFU	DCOM	CCOM	
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Key Scan Register Summary Table

Name	Address	Description
P_KS_Ctrl	0x7BC0	Key Scan Control Register
P_KS_Data0	0x7BC8	Sample Data of Line IOA[0]
P_KS_Data1	0x7BC9	Sample Data of Line IOA[1]
P_KS_Data2	0x7BCA	Sample Data of Line IOA[2]
P_KS_Data3	0x7BCB	Sample Data of Line IOA[3]
P_KS_Data4	0x7BCC	Sample Data of Line IOA[4]
P_KS_Data5	0x7BCD	Sample Data of Line IOA[5]
P_KS_Data6	0x7BCE	Sample Data of Line IOA[6]
P_KS_Data7	0x7BCF	Sample Data of Line IOA[7]

P_KS_Ctrl		0x7BC0						Key Scan Control Register									
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	INT/C	IEN	AUTO	FIXSTIME	INV	SMART	STRSCAN	BY	STOP	B74OFF	B31OFF	B0OFF	STIME	TSEL			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

P_KS_Data0		0x7BC8						Sample Data of Line IOA[0]								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data0							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data1								0x7BC9								Sample Data of Line IOA[1]							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Function	-	-	-	-	-	-	-	-	Data1														
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							

P_KS_Data2			0x7BCA						Sample Data of Line IOA[2]							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data2							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data3			0x7BCB						Sample Data of Line IOA[3]							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data3							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data4				0x7BCC					Sample Data of Line IOA[4]							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	-	-	-	-	-	-	-	-	Data4							
Default	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data5		0x7BCD								Sample Data of Line IOA[5]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data5							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data6		0x7BCE								Sample Data of Line IOA[6]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data6							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_KS_Data7		0x7BCF								Sample Data of Line IOA[7]							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		-	-	-	-	-	-	-	-	Data7							
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Miscellaneous Register Summary Table

Name	Address	Description
P_Byte_Swap	0x7BD0	Byte Swap
P_Nibble_Swap	0x7BD1	Nibble Swap
P_TwoBit_Swap	0x7BD2	Two-Bit Swap
P_Bit_Reverse	0x7BD3	Bit Reverse

P_Byte_Swap		0x7BD0								Byte Swap							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Nibble_Swap		0x7BD1								Nibble Swap							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_TwoBit_Swap		0x7BD2								2-Bit Swap							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

P_Bit_Reverse		0x7BD3								Bit Reverse							
Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function		B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0
Default		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

E-Fuse Register Summary Table

Name	Address	Description
P_EFuse_D0	0x7C30	E-Fuse Data Register 0
P_EFuse_D1	0x7C31	E-Fuse Data Register 1
P_EFuse_D2	0x7C32	E-Fuse Data Register 2
P_EFuse_D3	0x7C33	E-Fuse Data Register 3

P_EFuse_D0		0x7C30						E-Fuse Data Register 0								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [15:0]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

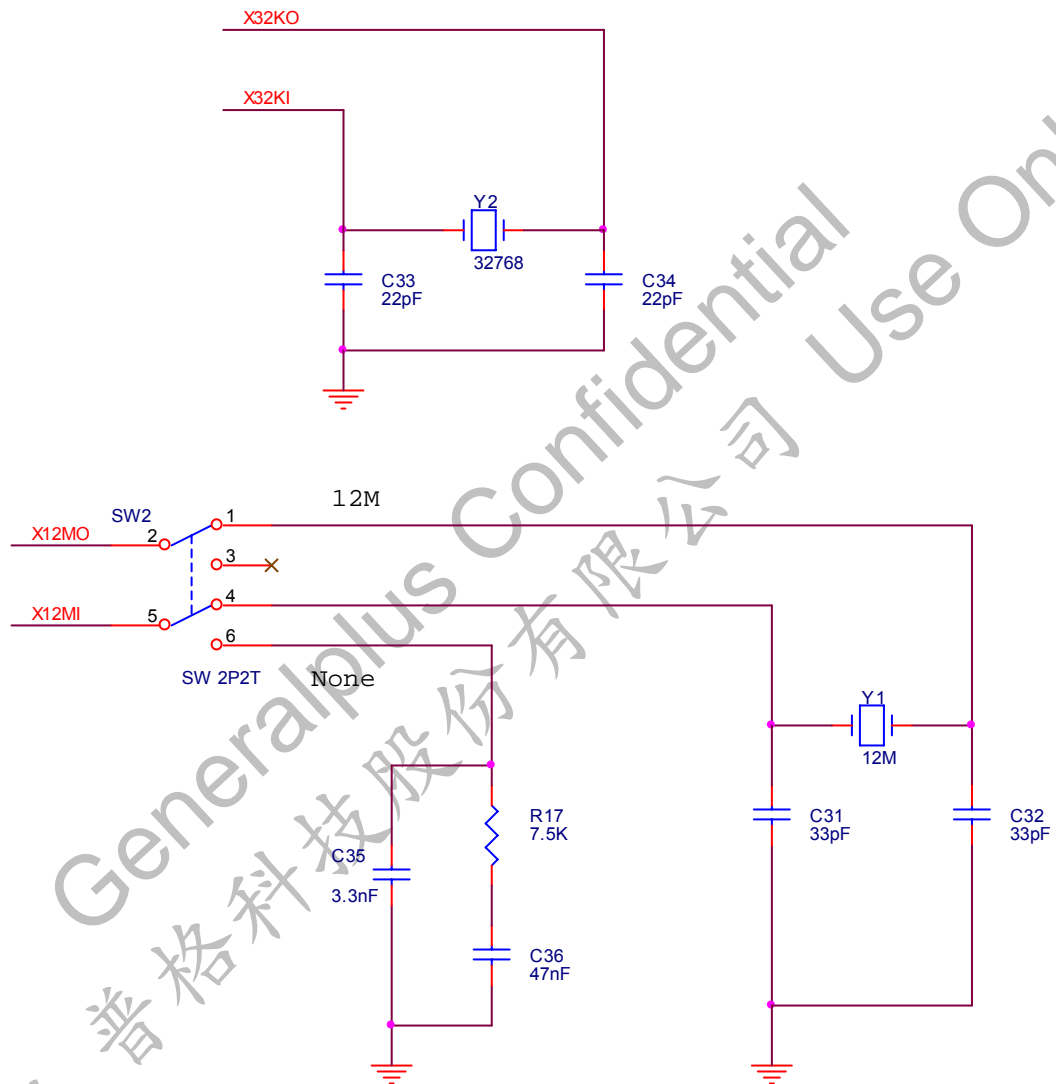
P_EFuse_D1		0x7C31				E-Fuse Data Register 1										
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Function	E-DATA [31:16]															
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

P_EFuse_D2		0x7C32								E-Fuse Data Register 2							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	E-DATA [47:32]																
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

P_EFuse_D3		0x7C33		E-Fuse Data Register 3													
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Function	E-DATA [63:48]																
Default	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

25.3 Crystal Usage Guide

The following components must be placed as closed as possible to the GPL162002A/162003A while designing layout and the value of components should not be changed.



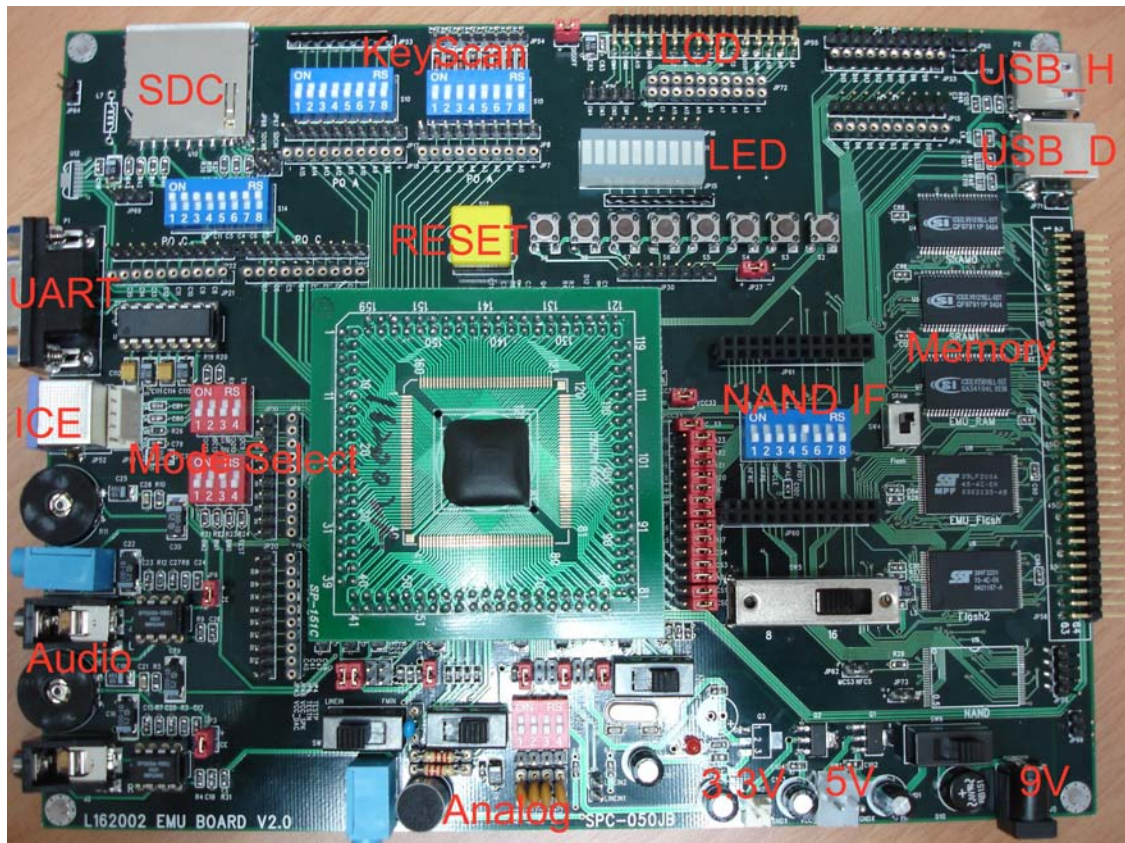
25.4 Development (Emulation) Board Configuration

This appendix gives a brief overview on GPL162002A/162003A development (emulation) board and gives details on how to use this board to emulate varied products.

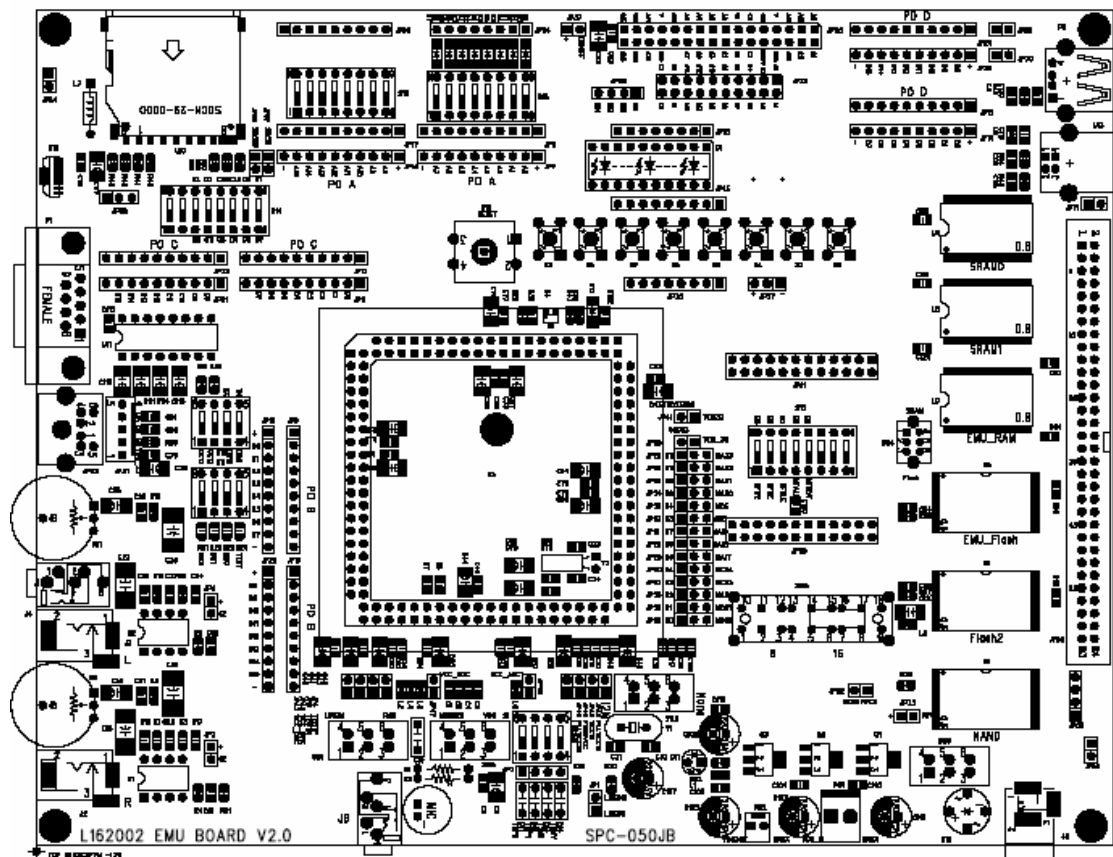
I. Connection



II. Emulation Board



III. Emulation Board Top View



25.4.1 Power Adjustment & Selection

There are three ways to supply DC power for GPL162002A/162003A EMU Board. If the board is powered, LED (indicated as D11) will be turn on.

- PW1: Direct power input, bypass 5V regulator.
- PW2: Direct power input, bypass 3.3V regulator and 5V regulator. When this mode is used, the USB power 5V is not provided.
- J5: Adapter DC input, on board regulators (5V and 3.3V) are activated. Input voltage should higher than 6V.

25.4.2 Reset

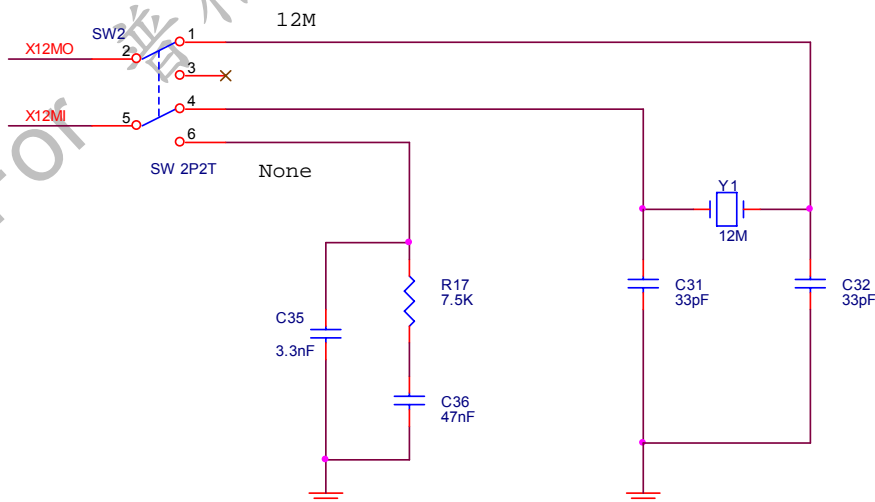
The switch S12 (labeled with **RESET**) resets the system manually and initializes GPL162002A/162003A emulation board.

25.4.3 ICE

JP51, JP52 (labeled with **ICE**) are the connectors for attaching ICE Probe.

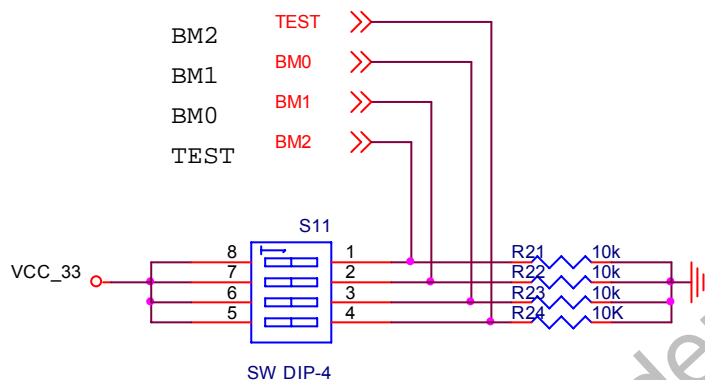
25.4.4 12MHz Crystal

In GPL162002A/162003A, users can select to use external 12MHz crystal or not. When the external 12MHz crystal is used, users should turn on the DIP switch 1 on S11 and switch SW2 to "12M" side. When the external 12MHz crystal is not used, users should turn off the DIP switch 1 on S11 and switch SW2 to "None" side. Generalplus suggests that when the built-in USB function GPL162002A/162003A is used, users should enable the external 12MHz crystal.



25.4.5 CPU Boot mode

GPL162002A/162003A include three boot modes that described in **MEMORY** chapter. The DIP switch 2, 3 on S11 are to select among these three boot modes.



25.4.6 Memory

There are six memory footprints on emulation board: three for SRAM, two for NOR flash memory and one for NAND flash memory. Since GPL162002A/162003A has a built-in 128KW mask ROM, programmers can use external memory devices to simulate internal mask ROM. By the switch SW4, users can select SRAM or NOR-type flash memory to simulate internal mask ROM.

GPL162002A/162003A allows the memory control signals CS0 ~ CS4, MA17 ~ MA23 and OEB, WEB to be set as GPIO. When ProtD is configured as GPIO, Jumper J25, J26, J28, J29, J31, J32, J34, J35, J36, J37, J39 and J40 must be properly set up.

25.4.7 Audio Output

J2 and J3 are single channel phone-jacks to external speakers. On GPL162002A/162003A EMU board, these two audio outputs are amplified by Generalplus OP amplifiers, known as GPY0030. Users can modify the gain of the amplifiers for CHA and CHB by simply changing the value of corresponding potential resistor (R6 and R11).

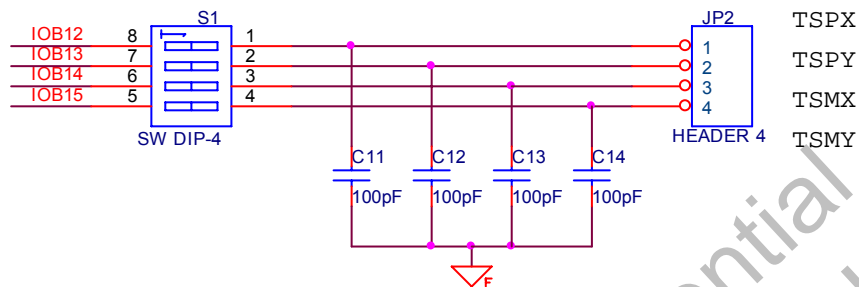
J4 is the stereo phone-jack for an external headphone. In this situation, GPL162002A/162003A drives the headphone directly.

25.4.8 Analog Input

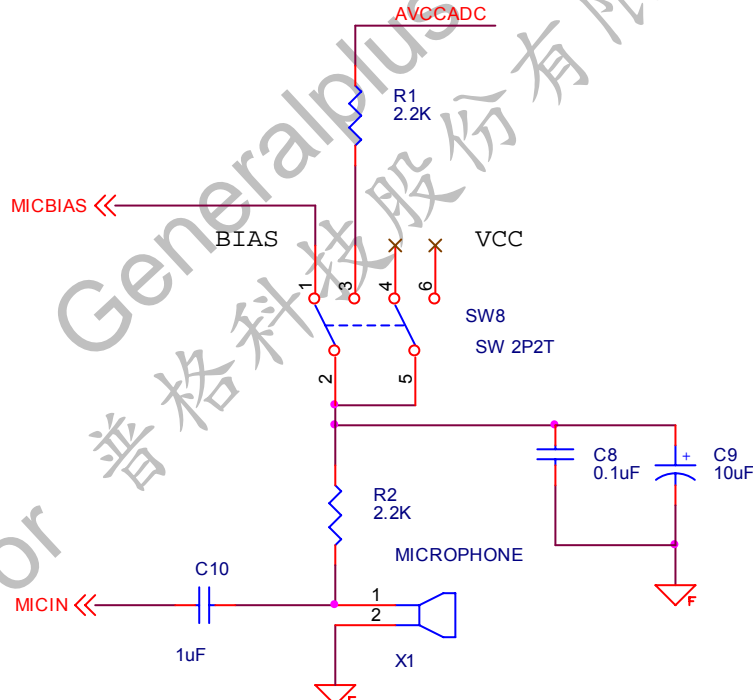
JP2 is the header for touch panel interface with 4 bypass capacitors to ground. Note that if touch panel

interface is used, turn on corresponding switches on S1 which interconnect GPL162002A/162003A and PortB socket (Referenced as JP19 and JP20, labeled with **IOB**).

JP1 is the header for 2 general-purposed analog inputs (line1 and Line2), and users can connect external device to internal ADC via this header. Note that Line1 and Line2 are dedicated pins of GPL162002A/162003A, and Line3, Line4 are shared with GPIO.



SW8 is used to select microphone bias voltage provided from VDD power or from GPL162002A/162003A MIC bias output. If GPL162002A/162003A MIC bias voltage is used, the MIC bias control bit (0x7970.b1) should be enabled.



SW1 is used to select AD recording input source. The selection should correspond to bit [5:4] of control register for HQADC input source selection.

25.4.9 Others Input and Output

I/O Ports:

PortA (Referenced as JP7, JP8, JP17 and JP18, labeled with **IOA**)

PortB (Referenced as JP9, JP10, JP19 and JP20, labeled with **IOB**)

PortC (Referenced as JP11, JP12, JP21 and JP22, labeled with **IOC**)

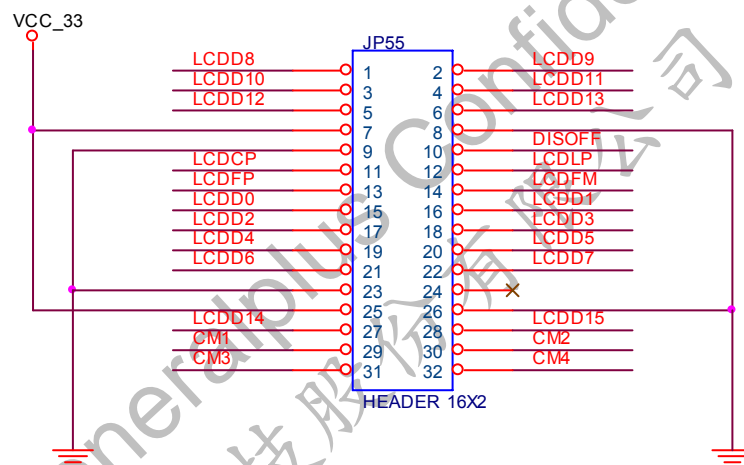
PortD (Referenced as JP13, JP14, JP23 and JP24, labeled with **IOD**)

LCD Driver Interface Header

Referenced as JP55 and JP72.

JP57, connected to LCDEN (DISPOFF) pin of external LCM, is used to enable the LCM. Shorting JP57.1 and JP57.2 will enable LCM all the time. In addition, users can connect one designated GPIO pin to JP57 to turn the LCM on or off.

Refer to the LCD schematic diagram for detail pin assignment.



UART/IrDA Interface Header

Referenced as JP69

Note that, if UART or IrDA interface is enabled, both TX (3) and RX (4) DIP switches on S16 should be turned on. Besides, GPL162002A/162003A emulation board also provides a RS232 level-shifter IC (HIN232) with female output header (referenced as P1) for UART.

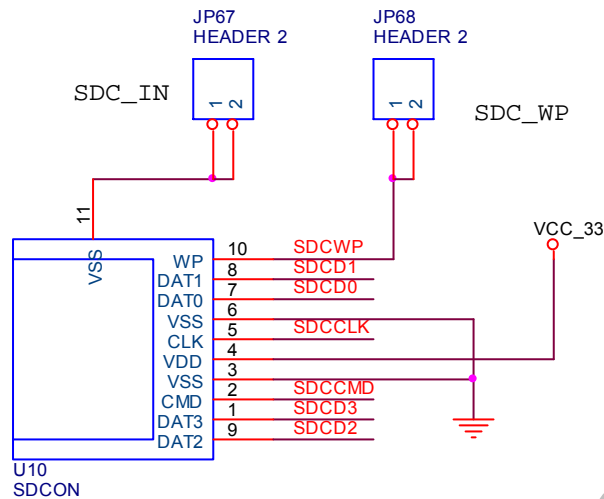
GPL162002A/162003A emulation board also reserves a footprint of an IrDA transceiver module (HP3201). This IC is optional, and not soldered on the PCB.

SD Card Interface Header

Referenced as U10

Note that if SD Card interface is used, turn on corresponding switches on S14 which interconnect GPL162002A/162003A and PortC socket (referenced as JP21 and JP22, labeled with **IOC**).

JP67 and JP68 are connected to SDC socket 11 and 10 pins respectively so that users can perform SDC writing protection and detecting insertion.



NAND Flash Interface Header

Referenced as U9

The built-in NAND Flash controller on GPL162002A/162003A supports 8-bit and 16-bit type by switching SW5 to select which type is used. When Nand Flash interface is used, users should turn on the DIP switch S13 which interconnects GPL162002A/162003A and PortB socket (referenced as JP9, JP10, JP19 and JP20, labeled with **IOB**).

JP73.2 is connected to NAND flash WP pin, and JP73.1 is connected to VCC. Shorting these two pins will disable NAND flash WP.

JP62.2 is connected to NAND flash CE pin, and JP62.1 is connected to MCS3. Shorting these bins will configure MCS3 as NAND type flash. In addition, for example, if users want to configure MCS2 as NAND type flash, JP62.2 and MCS2 pin of GPL162002A/162003A should be physically connected.

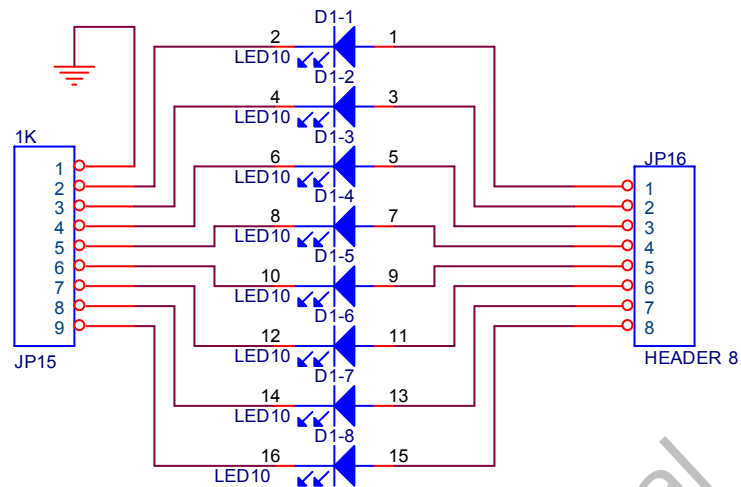
Key Scan Interface

When key scan function shares PortA with LCD panel, each output must connect a diode serially to the key pads to prevent the LCD glitch caused when multi keys are pressed. Users can turn on DIP switches S15 and S10 to achieve this goal.

On-board LED indicator Header

Referenced as JP16

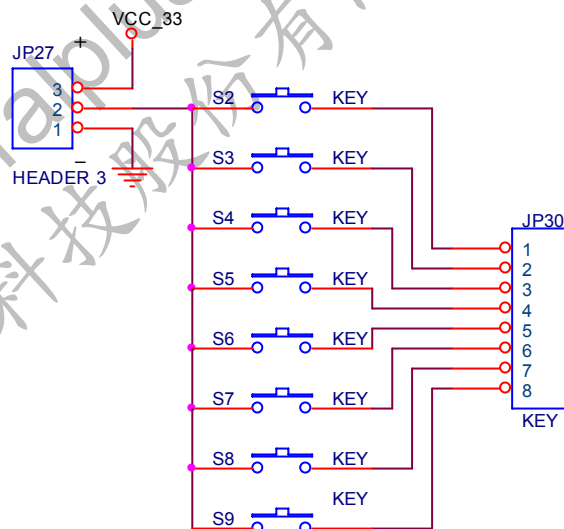
This LED array is for general-purposed indicator(s), and connecting any one pin on JP16 to high voltage will turn on corresponding LED.



On-board Key Array Header

Referenced as JP30

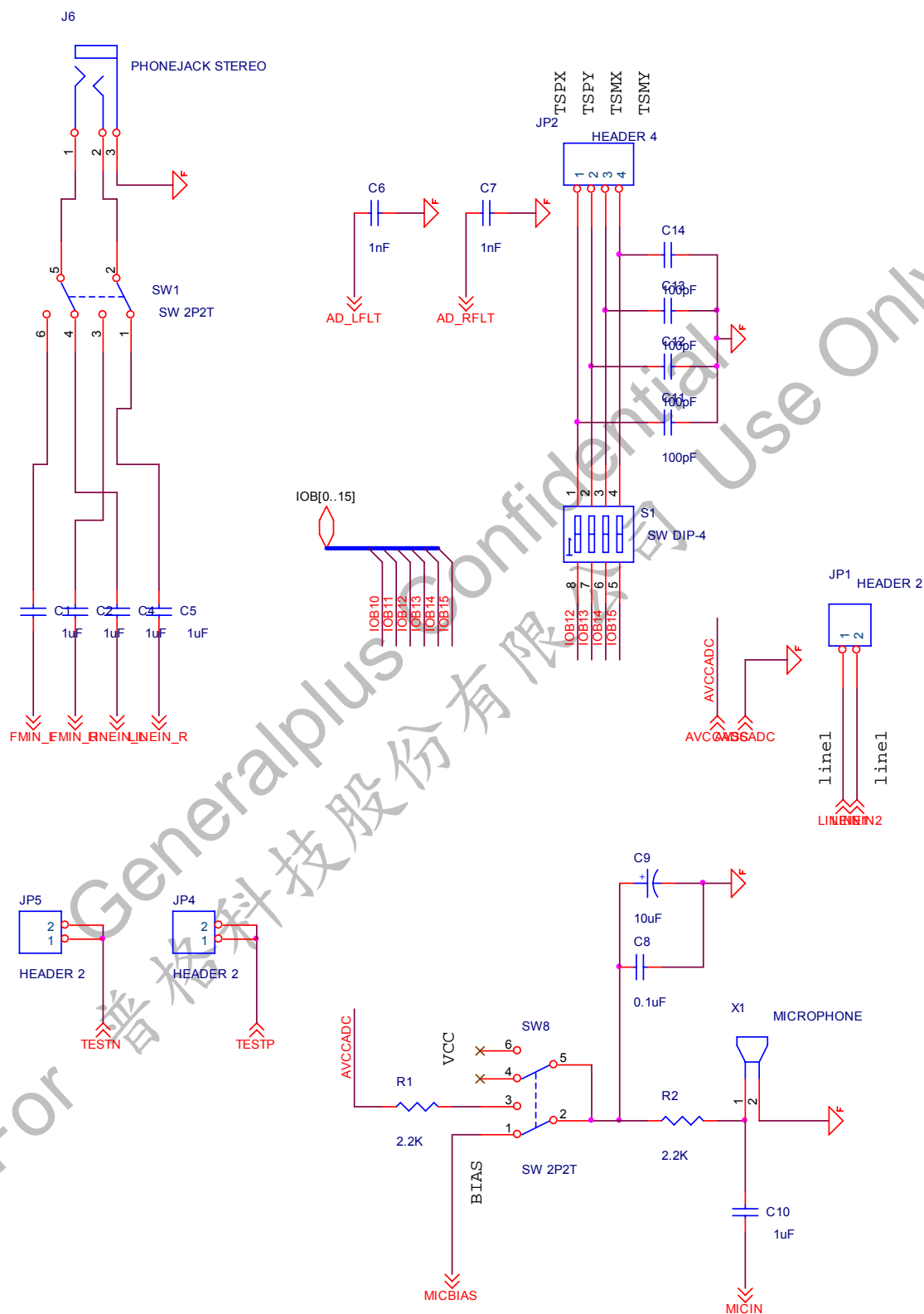
This key array is for general-purposed input(s). Each pin on JP30 connects to each corresponding key individually and then to the one common pin in the middle of JP27. This common pin can be easily shorted to high voltage or low voltage via a single jumper.



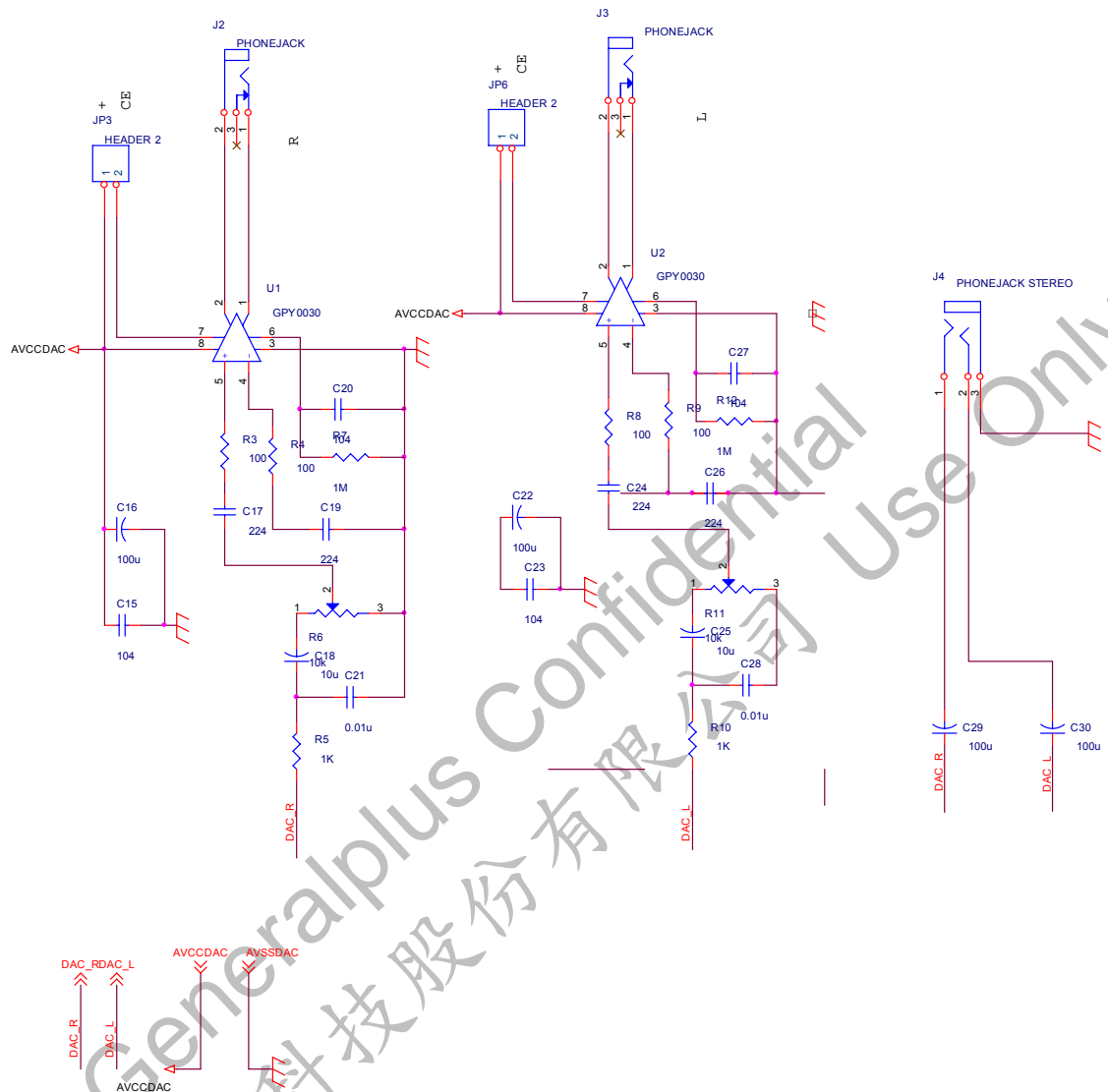
25.5 Development (Emulation) Board Schematic

- A. Analog Input
- B. Audio Output
- C. Clock
- D. GPL162002A/162003A
- E. GPIO
- F. ICE
- G. UART/IrDA
- H. Key Scan
- I. LCD
- J. Memory
- K. Nand Flash
- L. Power
- M. SDC
- N. USB

A. Analog input

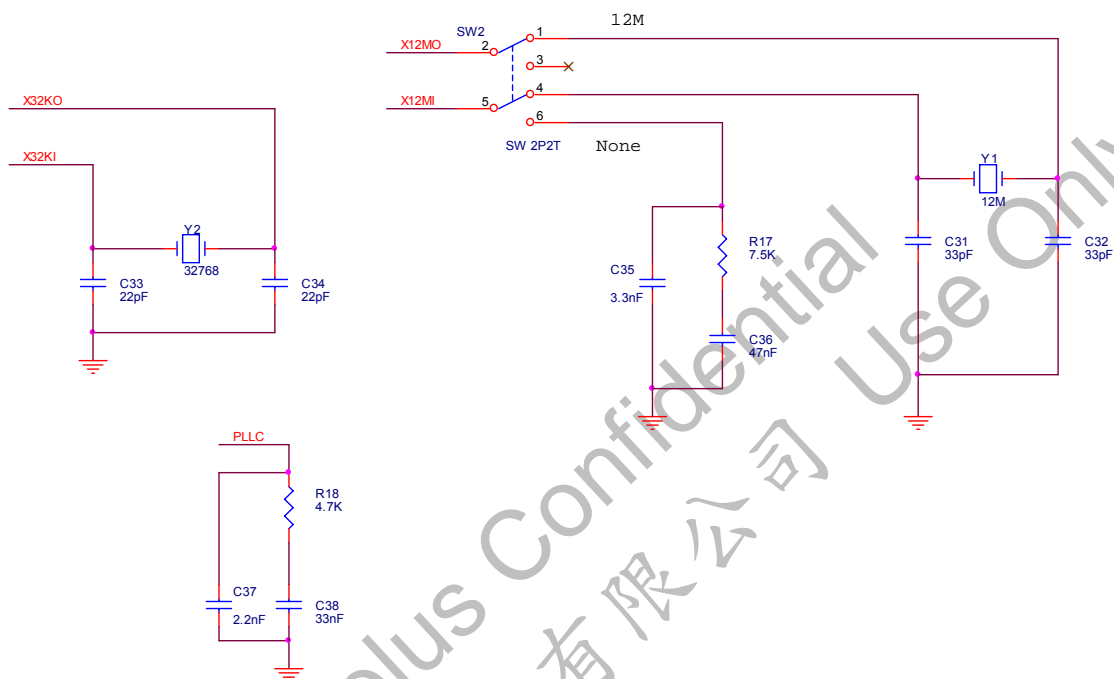


B. Audio output

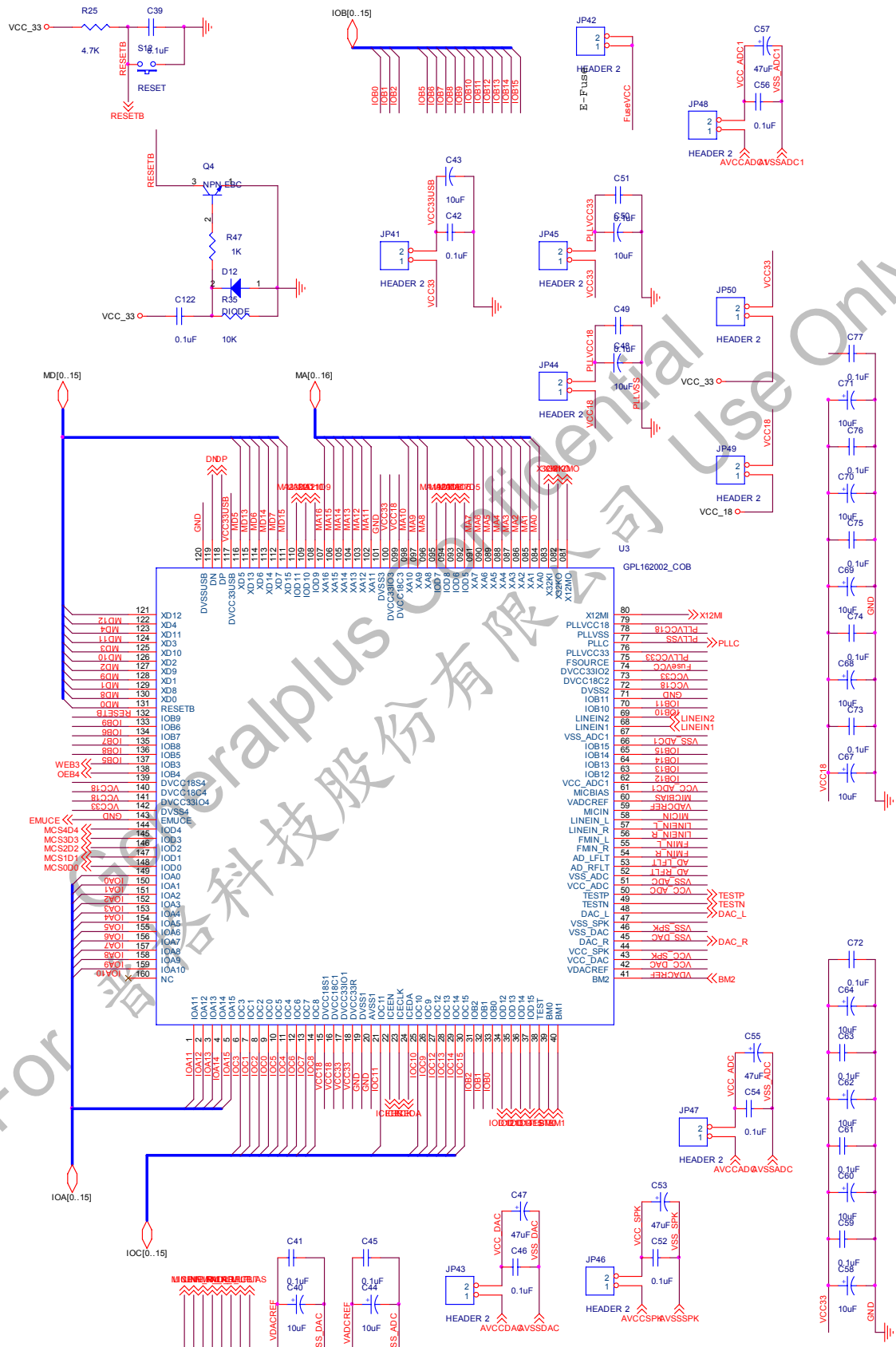


C. Clock

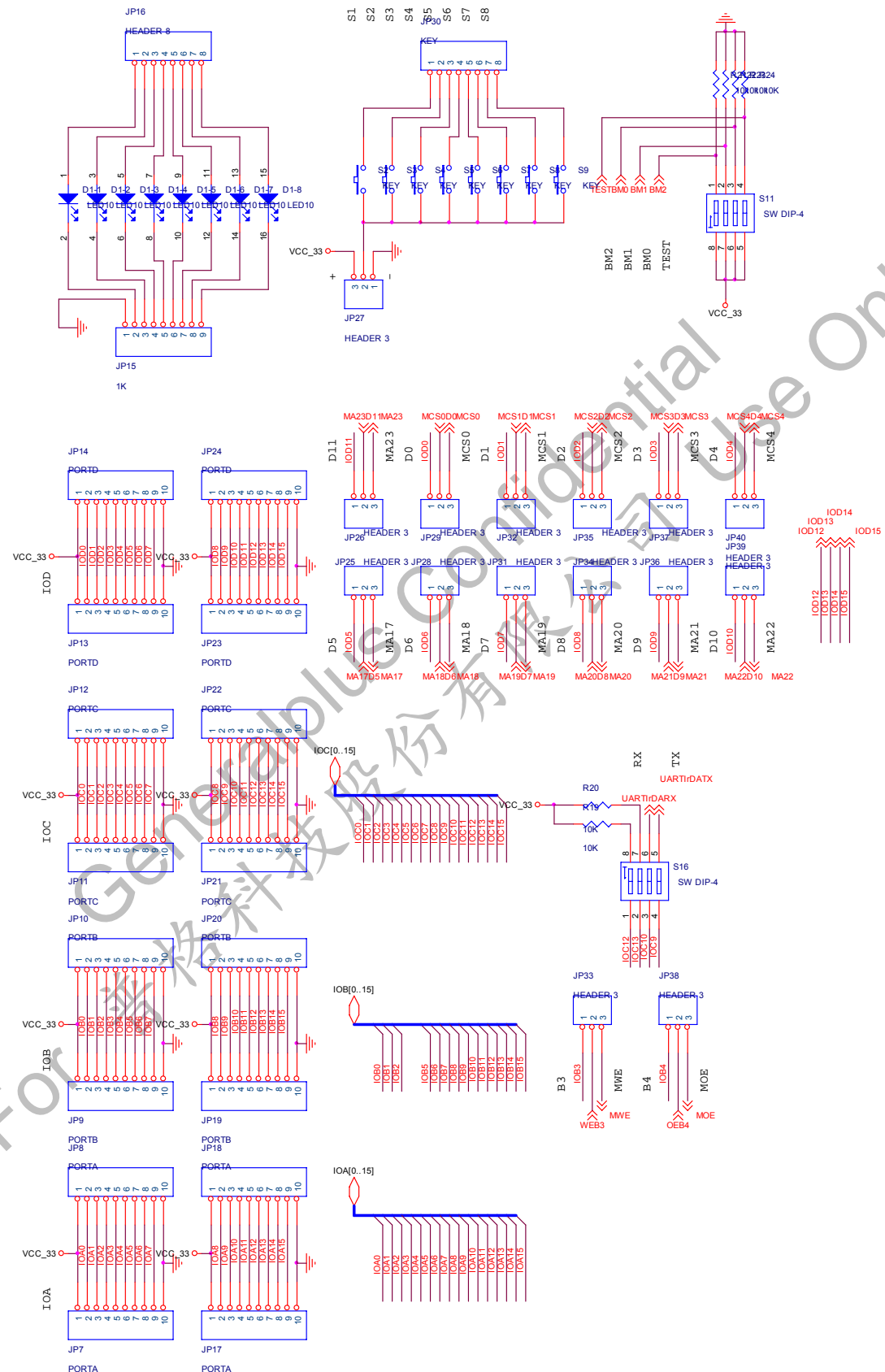
X32KI >> X32KI
 X32KO >> X32KO
 X12MO >> X12MO
 X12MI >> X12MI
 PLLC >> PLLC



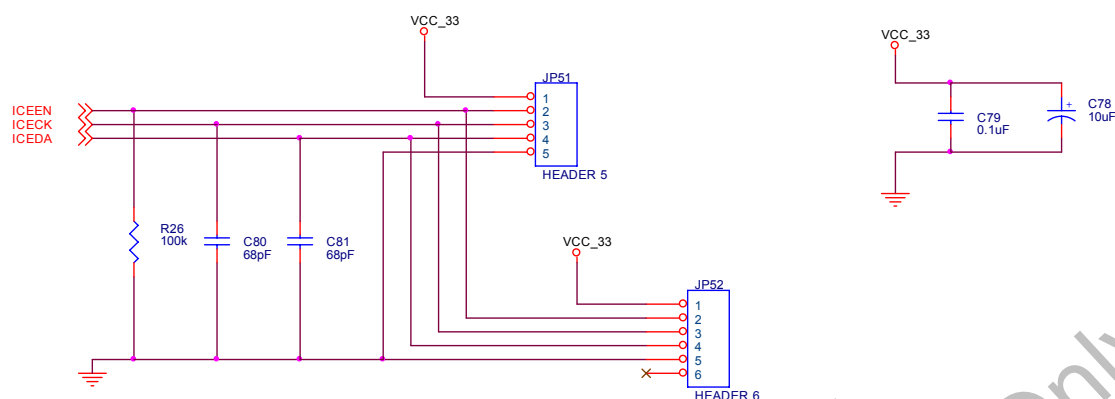
D. GPL162002A/162003A



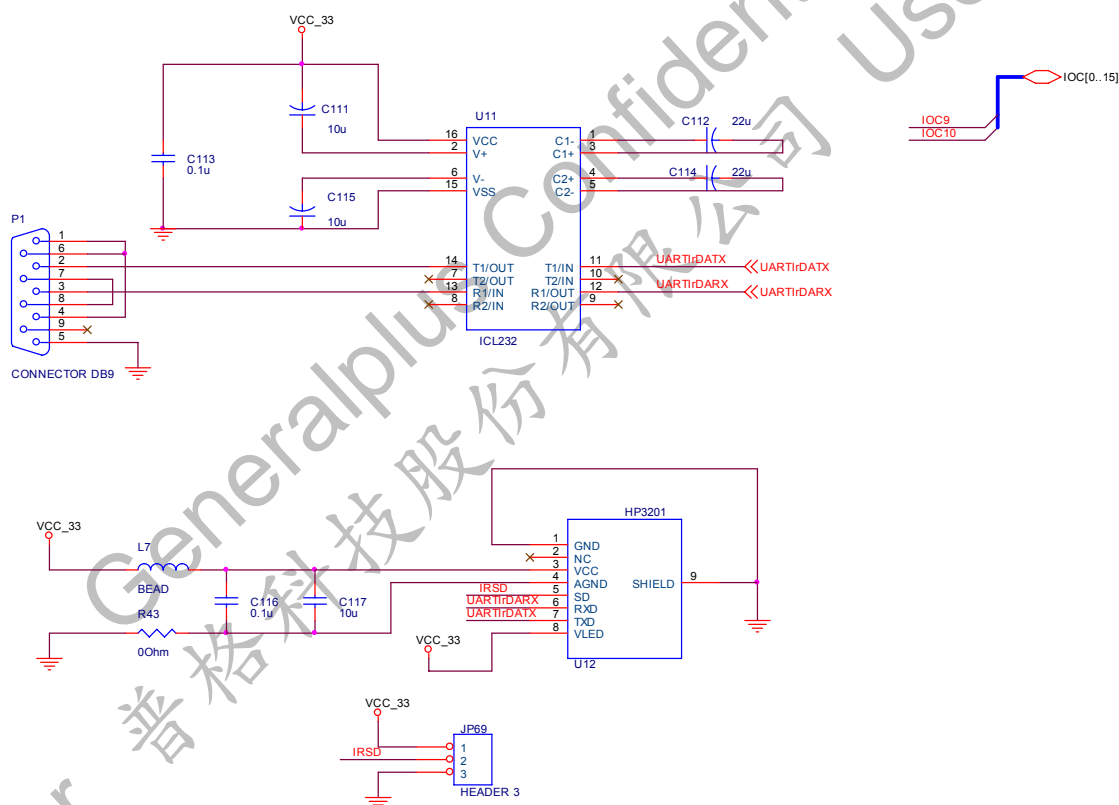
E. GPIO



F. ICE

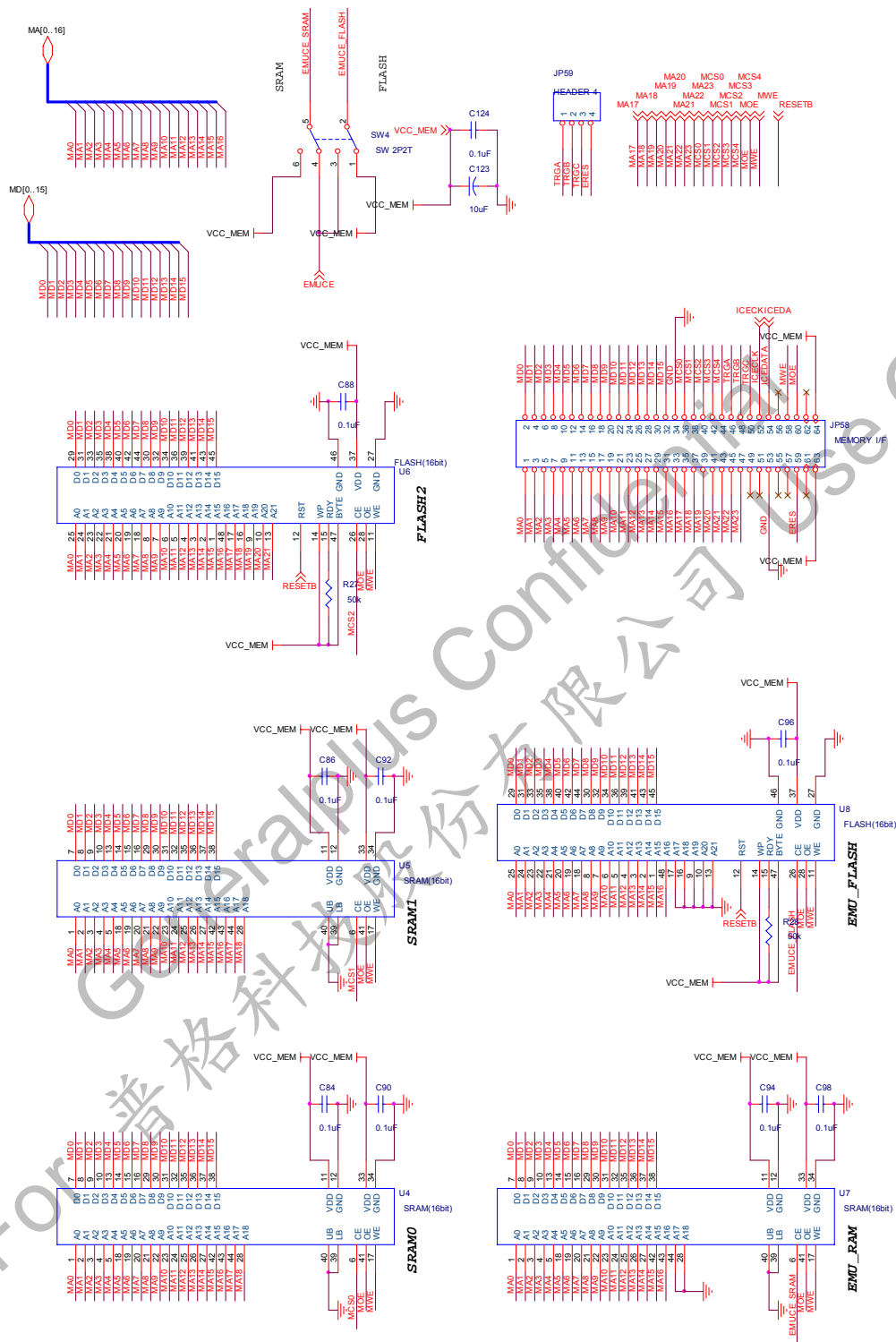


G. UART/IrDA

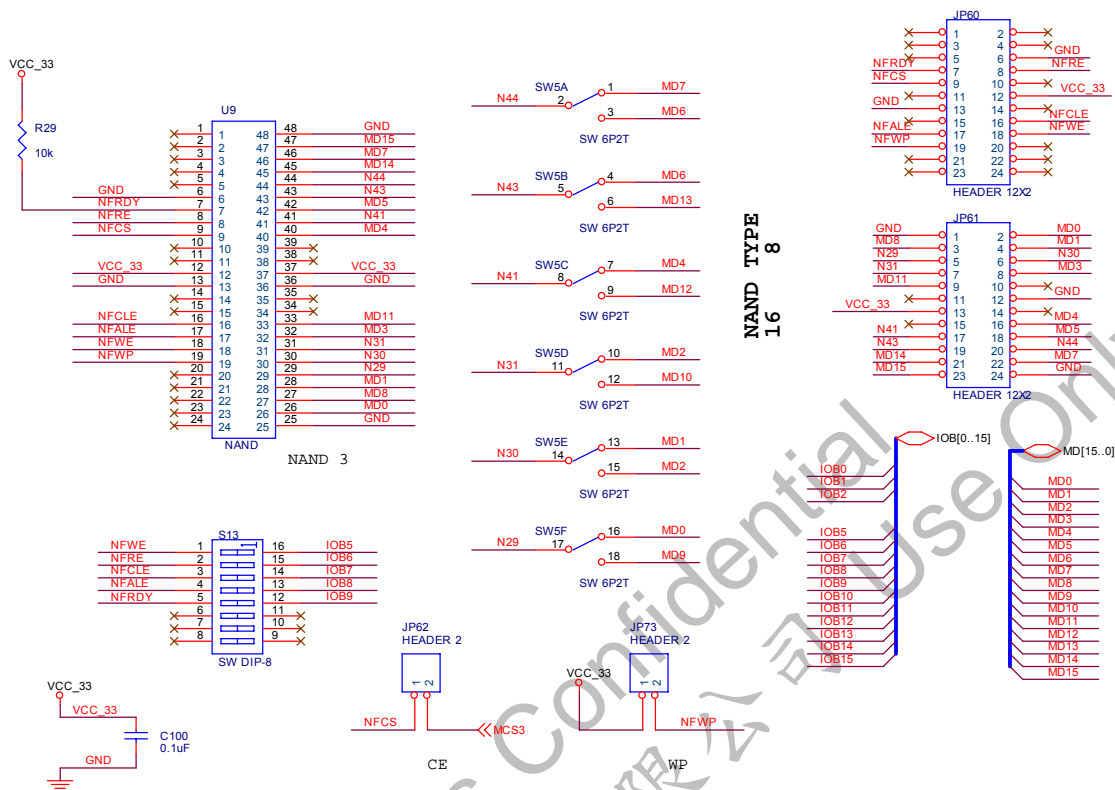


The diagram illustrates a 16-channel diode switch circuit. It features a 74VHC151 decoder (JP53) with 4-bit address inputs (A0-A3) and 16 outputs (Y0-Y15). The outputs are connected to a 16-pin DIP switch (S10) and a 16-pin header (JP54). The switch is connected to a 1k resistor and ground. The 16 channels are labeled IOA0 through IOA15.

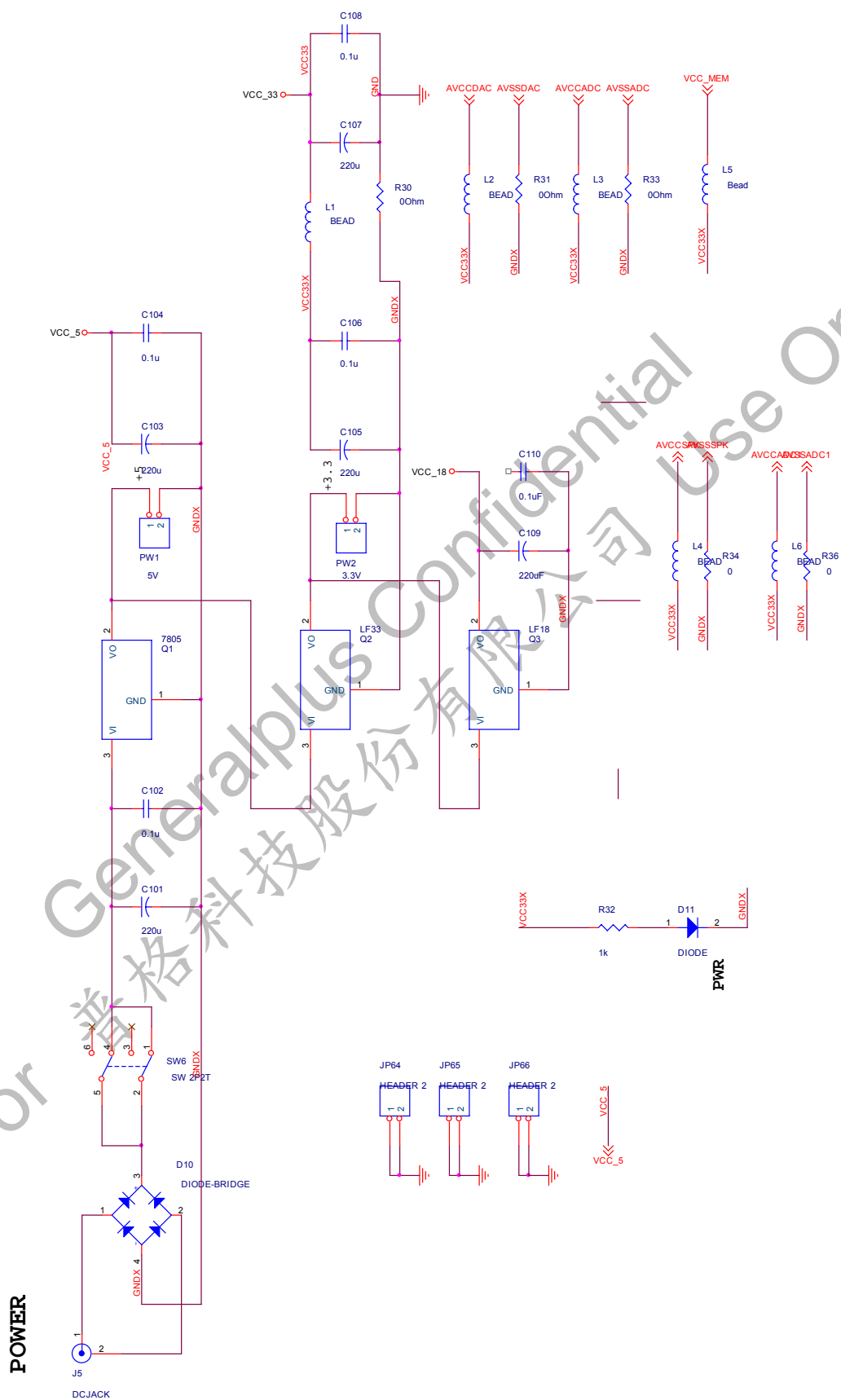
J. Memory



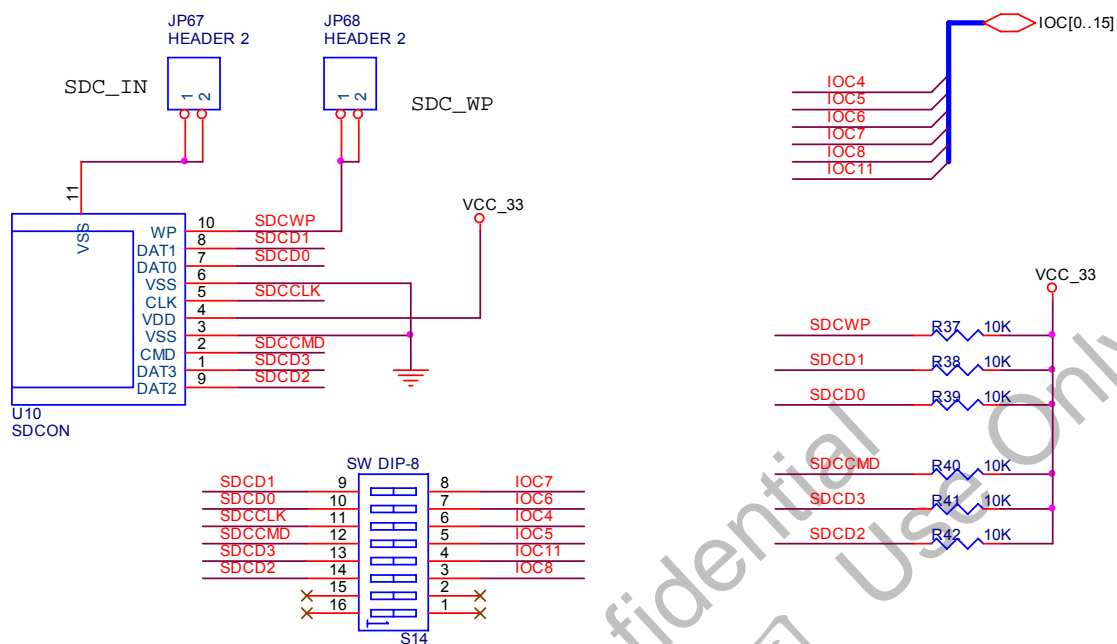
K. Nand Flash



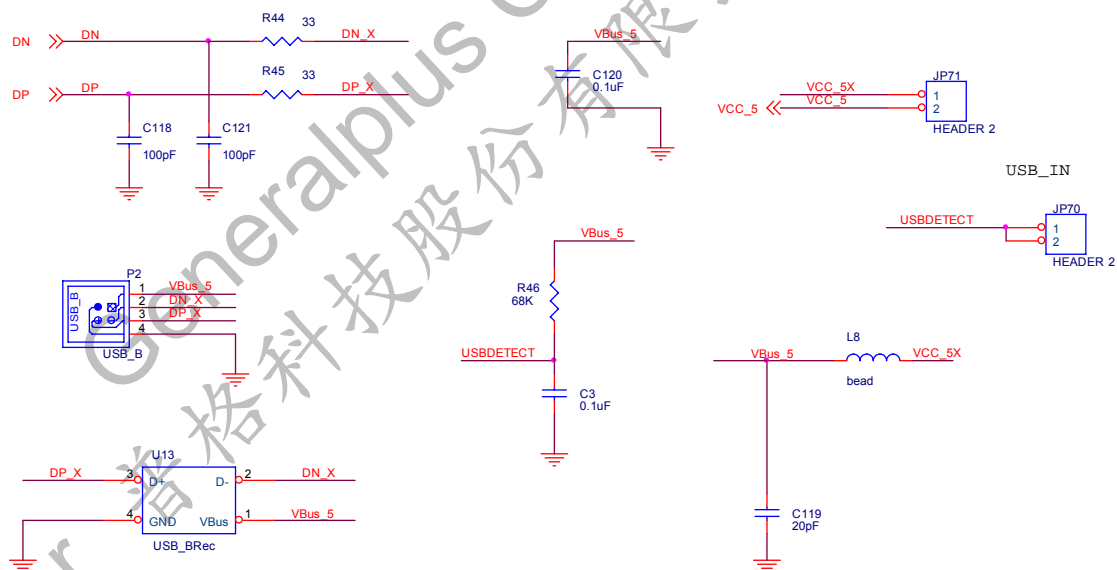
L. Power



M. SDC



N. USB



25.6 CPU Performance Downgrade Issue

25.6.1 LCD Display and System Performance

GPL162002A/162003A can use internal SRAM or external memory as display buffer. If LCD buffer is configured as an external memory device, (the display data should be read from external memory) and running program is on the other external memory device, these two memory devices will share the same system bus. Therefore, CPU performance is degraded.

As depicted in the following table, there are three memory-access conditions on GPL162002A/162003A. When LCD and CPU access the same region of memory, the CPU performance of CPU will be downgraded. At this situation, Generalplus recommends take the following action to improve the CPU performance:

Copy the instruction into SRAM, and execute it in internal SRAM. Or, define the LCD buffer in the internal SRAM.

	LCD uses internal SRAM as data buffer	LCD uses external memory as data buffer
Software Program On Internal RAM	downgrade CPU performance	Do not downgrade CPU performance
Software Program On Internal ROM	downgrade CPU performance	Do not downgrade CPU performance
Software Program On External memory	Do not downgrade CPU performance	downgrade CPU performance

If it is difficult to take above actions, users should take care of the bandwidth of LCD occupying data bus. The factors include the wait cycle for external memory, the color/gray mode, the size of LCD panel, and the Frame rate of color/gray display in LCD panel.

The formula of downgrade factors:

The percentage that STN/TFT LCD interface occupies bus bandwidth equals to

$$\text{LCD_Segment} \times \text{LCD_Common} \times \text{LCD_BPP} \times \text{LCD_Frame_Rate} \times \text{LCD_Wait} / 16 / \text{PLL_Clock}$$

where

LCD_BPP = display mode = 1, 2, 4, 8, 12, or 16 bit per pixel

LCD_Wait = extra wait cycle of LCD buffer for enough external memory accessing time

This value is the same with the value in P_MCSx_Ctrl

CSx depends on which LCD buffer is located

If the LCD buffer is located in internal SRAM, then LCD_Wait = 1

PLL clock = depending on the value set in P_Clock_Ctrl

LCD_Frame_Rate = the enough frequency for color/gray display

For example:

PLL clock = 48MHZ

LCD_Wait= 3 (It means the LCD Buffer access time is less than 3 PLL clock cycle)

X-size x Y-size x Frame rate	BPP=1	BPP=2	BPP=4	BPP=8	BPP=12	BPP=16
320 * 240 * 156.25	4.6875%	9.375%	18.75%	37.5%	56.25%	75%
320 * 240 * 125	3.75%	7.5%	15%	30%	45%	60%
320 * 240 * 60	1.8%	3.6%	7.2%	17.4%	21.6%	34.8%
160 * 160 * 125	1.25%	2.5%	5%	10%	15%	20%

25.7 Audio Output Components Selection Guide

DAC Output RC circuit vs. frequency response

For DAC audio output, the external audio driver circuit is required. The bandwidth of the audio driver circuit can affect the perceived audio performance. User can adjust the bandwidth according to the reference table below. R is the effective resistance of the circuit and C is the effective capacitance.

Please refer to GPL162002A/162003A application circuit.

F _{3dB} (3dB frequency)		R (ohm)															
		100	120	180	220	330	390	470	510	560	680	750	820	910	1000	1100	1500
C(uF)	0.0047	338.628	282.19	188.127	153.922	102.614	86.8276	72.0485	66.3976	60.4693	49.7982	45.1504	41.2961	37.2118	33.8628	30.7843	22.5752
	0.01	159.155	132.629	88.4195	72.3432	48.2288	40.809	33.8628	31.2069	28.4205	23.4052	21.2207	19.4092	17.4896	15.9155	14.4686	10.6103
	0.02	79.5775	66.3146	36.1716	36.1716	24.1144	20.4045	15.6034	15.6034	14.2103	11.7026	9.70458	9.70458	8.74478	7.95775	5.30517	5.30517
	0.047	33.8628	28.219	18.8127	15.3922	10.2614	8.68276	7.20485	6.63976	6.04693	4.97982	4.51504	4.12961	3.72118	3.38628	3.07843	2.25752
	0.1	15.9155	13.2629	8.84195	7.23432	4.82288	4.0809	3.38628	3.12069	2.84205	2.34052	2.12207	1.94092	1.74896	1.59155	1.44686	1.06103
	0.22	7.23432	6.0286	4.01907	3.28833	2.19222	1.85495	1.53922	1.41849	1.29184	1.06387	0.96458	0.88223	0.79498	0.72343	0.65767	0.48229
	0.47	3.38628	2.8219	1.88127	1.53922	1.02614	0.86828	0.72048	0.66398	0.60469	0.49798	0.4515	0.41296	0.37212	0.33863	0.30784	0.22575
	1	1.59155	1.32629	0.88419	0.72343	0.48229	0.40809	0.33863	0.31207	0.28421	0.23405	0.21221	0.19409	0.1749	0.15916	0.14469	0.1061

The frequency range, 5K~4 KHz is for speech application and marked in yellow cells.

The frequency range, 15K~25K Hz is for Audio application and marked in blue cells..

25.8 32768 Crystal and PLL Power-on Stable Time

When in power on, GPL162002A/162003A will operate in 12MHz system clock. And it takes 2048 cycle of 32768Hz crystal to make slow PLL clock stable.

After the bit, Fast PLL Enable in P_Clock_Ctrl, is set to 1, the system clock will change to 48MHz. And it takes 2048 cycle of 12MHz crystal to make fast PLL clock stable.

25.9 Reset Type

CPU reset:

When watchdog timeout reset, watchdog mode protect reset, or power-saving mode protect reset occurs, it just resets CPU. The peripherals are not affected by the above reset mechanism, and keep in its original state. The corresponding reset flag can be read out to judge which reset happens.

System reset:

When Power-on reset, Low-Voltage reset, or watchdog timeout reset occurs, GPL162002 (including CPU and all peripheral) is reset to initial state excluding SRAM and Palette RAM. All the flags return to initial state including reset flag. So, the reset flags must be recorded to SRAM to avoid disappearance. In other words, CPU reset is the subset of system reset.

Watchdog timeout reset:

When Watchdog timeout reset is activated, the value of Axx5 must be written to Watchdog clear register within the period of users' definition. Otherwise, CPU reset occurs. Also note that reset target of watchdog-timeout can be selected between CPU and system by software (1-bit control register).

Watchdog mode protection reset:

When Watchdog timeout reset is activated, if the wrong value (not Axx5) is written to Watchdog clear register, CPU reset will occur.

Power saving mode protection reset:

GPL162002A/162003A define three power saving modes, Wait, Halt, and Standby mode. When entering power saving mode, the Wait, Halt, or Standby mode control register must be written the corresponding value. If the wrong value is written into the control register, CPU reset occurs.

Low voltage reset:

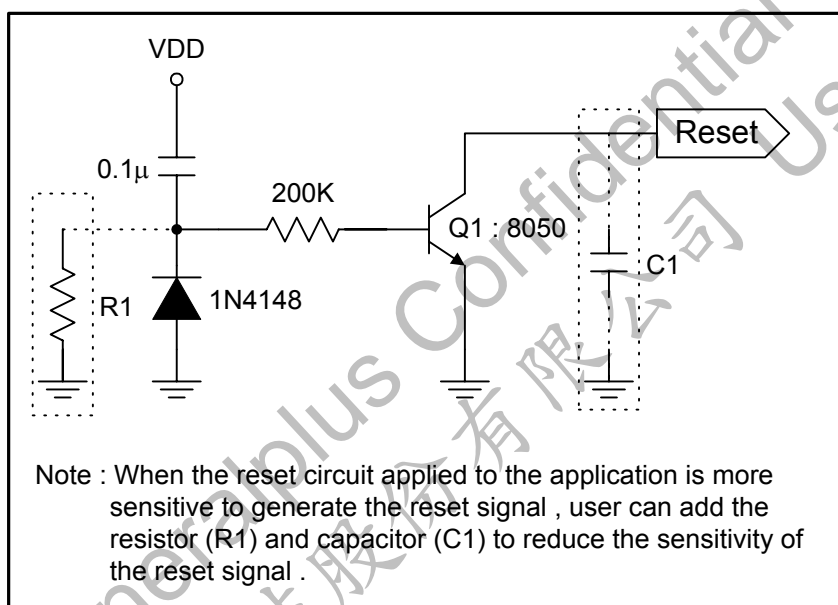
When the operation voltage is lower than 2.5V, the Low voltage reset mechanism will reset system.

Power on reset:

When power is supplied to GPL162002A/162003A from 0V, the power-on reset occurs. It has some probability that the power-on reset does not occur due to unstable operation voltage. The Super-reset mechanism is necessary to avoid this condition.

Super-reset circuitry:

The application circuit is shown below. It detects the variance of operating voltage and keeps the RESET signal as low state when operating voltage is charging from ground to stable state. When the operating voltage is in stable state, the Reset signal releases from '0' to '1'.



25.10 Important Note for the Setup of Memory Access Time

GPL162002A/162003A provides flexible adjustment of external memory access time. Through the adjustment of memory access time and flexible clock selection, users can fine-tune the system performance. For example, when PLL clock is 48MHZ, it means it has 20.8ns per clock cycle. In this case, when accessing the external memory, which has minimum access time 55ns, it takes 3 clock cycles to supply enough time to memory accessing. When PLL clock is 24MHZ, it means 41.7ns per clock cycle. So, it needs 2 clock cycles to supply enough accessing time.

There are six independent control registers to set the memory accessing time on GPL162002A/162003A, which are P_MCS0_Ctrl (0x7820), P_MCS1_Ctrl (0x7821), P_MCS2_Ctrl (0x7822), P_MCS3_Ctrl (0x7823), P_MCS4_Ctrl (0x7824), and P_EMUCS_Ctrl (0x7825).

When external memory devices are used, users must check the datasheet of memory to make sure the

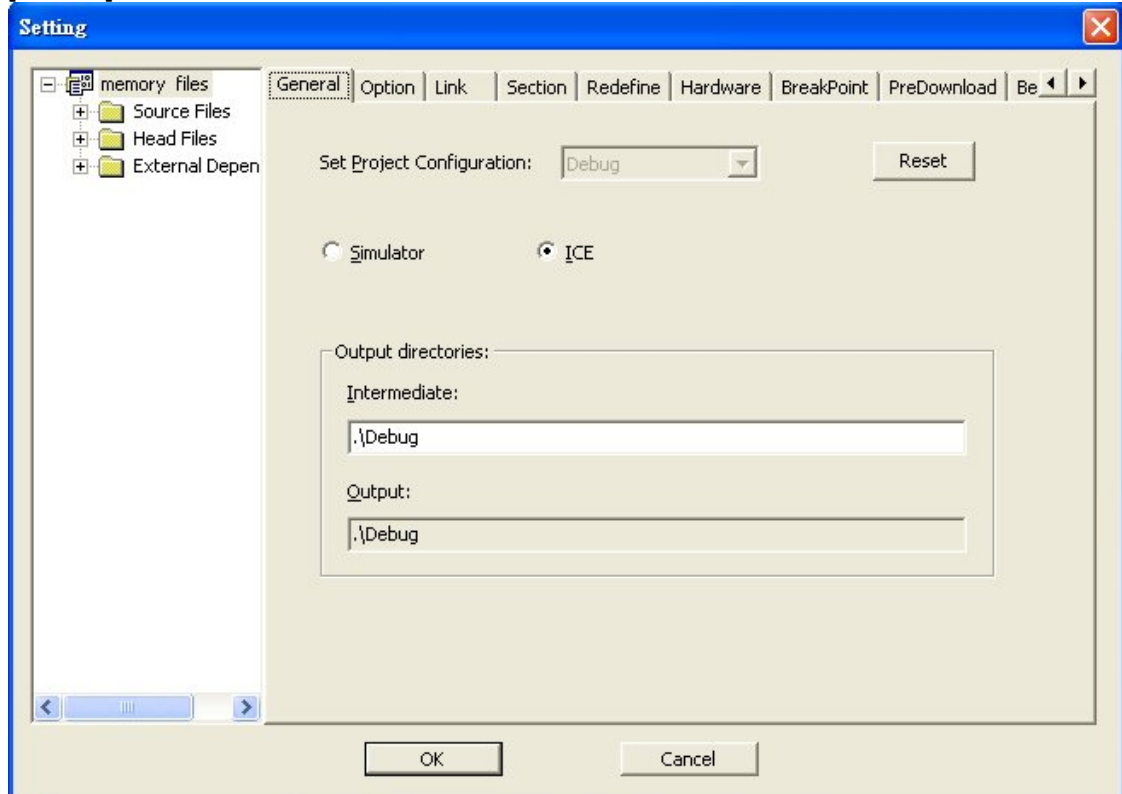
enough access time. The loading effect on PCB and operation voltage must be seriously considered.

25.11 Project Setting on IDE

For most projects, options are set at project level. Options can be setup for different objects that are created by either the system or a user. Project can display the structure of options for every object. The options given for the project level also apply to all files in the project.

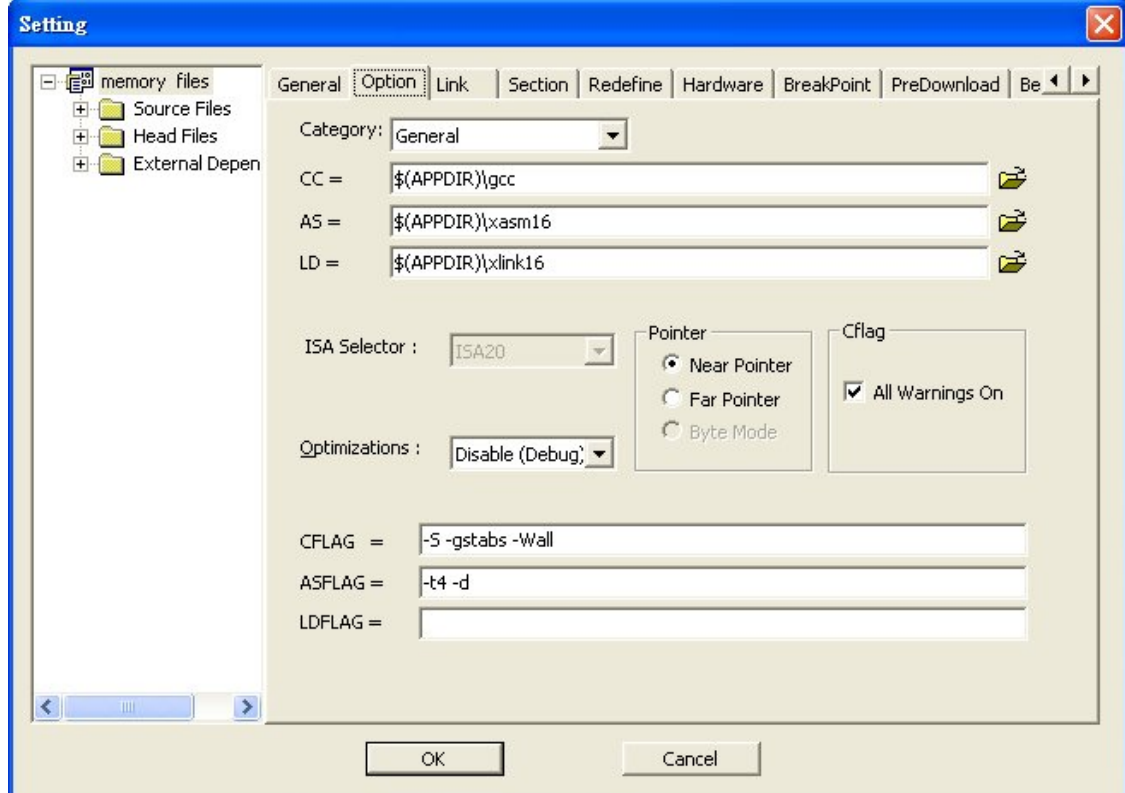
Set Option for Project

1. Open a project or create a project.
2. Click [Project]→[Setting] to display "Setting" dialog box.
3. Click on each label for further setting.

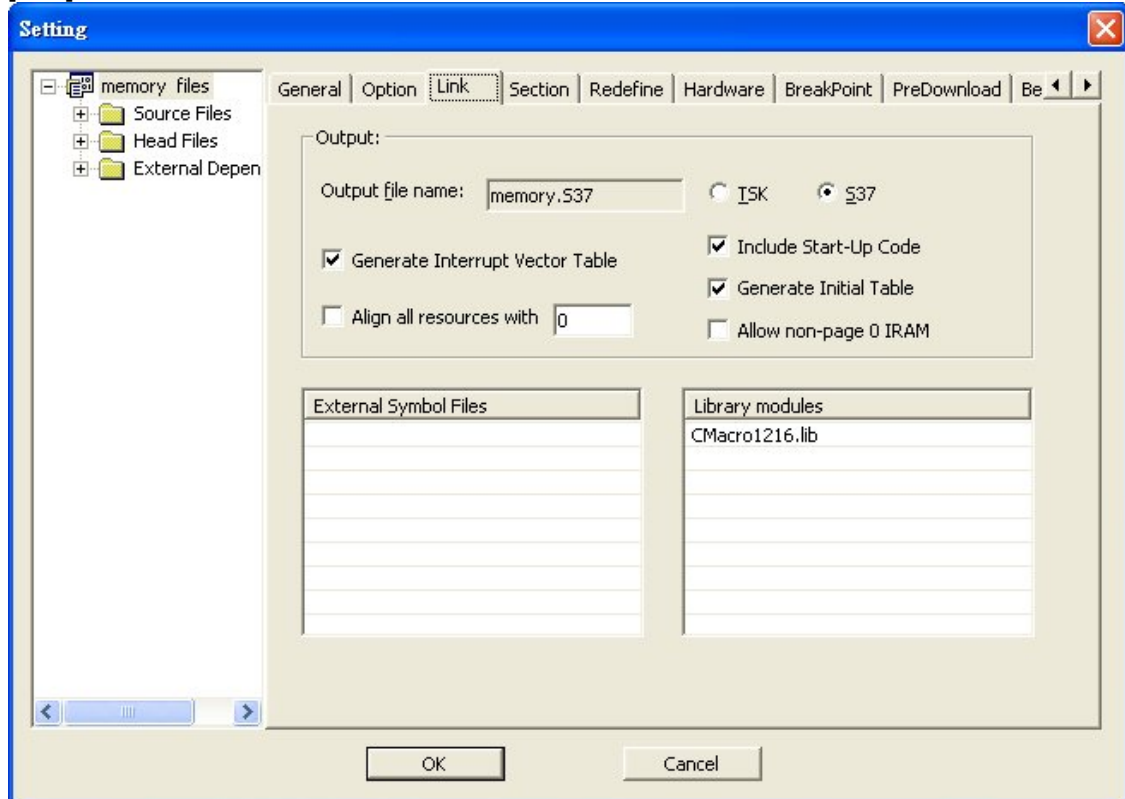
[General]


1. If simulator is chosen, all data will be stored into buffers.
2. Set Project Configuration: Select which version of the project is used, debug or release.
3. Simulator /ICE: select u'nSP® IDE running mode. If ICE mode is selected, you should connect an emulation board to your computer through a parallel port or a USB port.
4. PC trace enable: enable PC trace function (make sure the ICE board has PC trace capability during ICE mode).
5. Save instruction only: If checked only instruction fetch will be saved in PC trace buffer, else it will save all memory read/write in PC trace buffer.
6. Tracer buffer size: Specify the number of bytes to store operation records.
7. Intermediate: Specify directory for intermediate files. Intermediate files normally are generated during compilation.
8. Output: Display the terminal file's directory. Normally, it is the same as the intermediate directory.
9. Reset: Reset default configuration.

[Option]

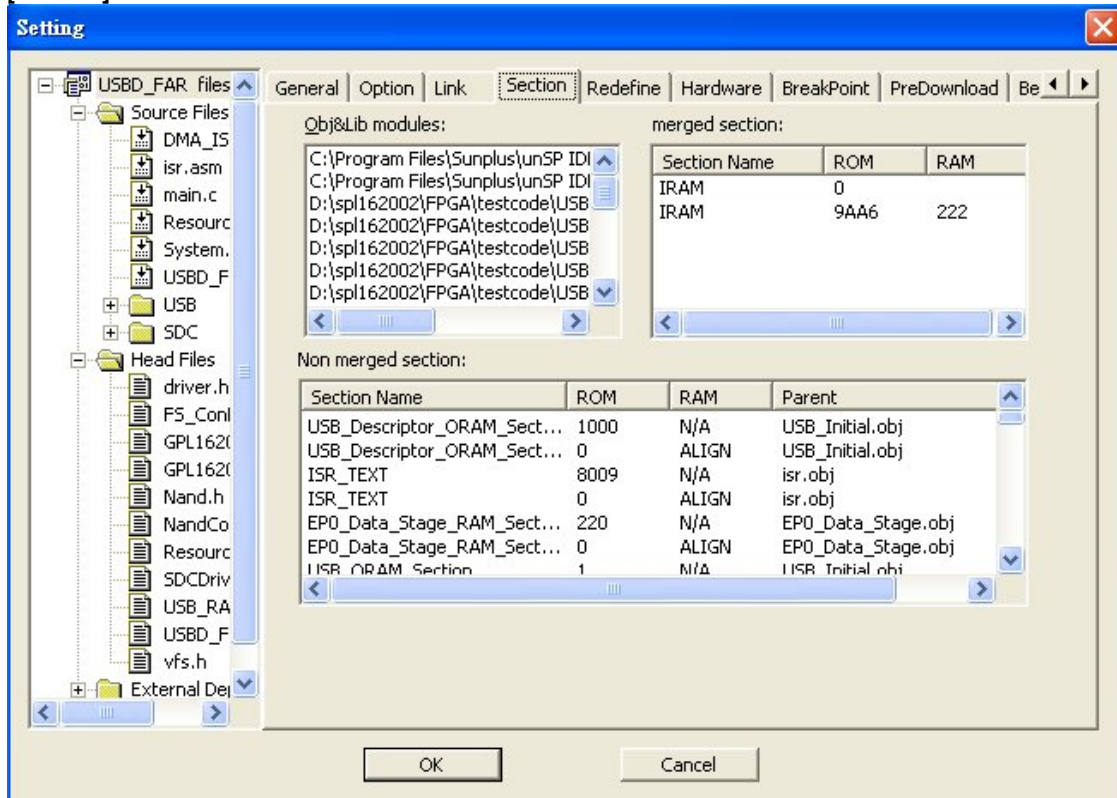


1. CC: Specify the C Compiler location.
2. AS: Specify the Assembler location.
3. LD: Specify the linker location.
4. CFLAG: Specify the C Compiler Operation FLAGS.
5. ASFLAG: Specify the assembler Operation FLAGS.
6. LDFLAG: Specify the linker Operation FLAGS.
7. Optimizations: Select the Optimization Type you want, the optimization flags will be changed automatically.
8. ISA Selector: Select different instruction set of unSP (ISA1.0, ISA1.1, ISA1.2 etc).
9. Makefile: Check if auto updating the makefile.
10. Ary file: Check if auto updating the array file.
11. Additional include dir: Set include file directory.
12. Additional library dir: Set library file directory.

[Link]


1. Output file name: Specify the output file's name.
TSK/S37: Types of object file (binary/ASCII (Motorola S37)). Before users define the type of output file, users should <check> both Makefile and Ary file at Option TAB.
2. Generate Interrupt Vector Table:
Uncheck it, if users don't want to produce Interrupt Vector Table in the project output.
3. Include Start-Up Code:
Uncheck it, if users don't want to produce the default start-up code in the project output.
4. Align all resource with:
Check it and input the align base, if users want to align all resources with a specified align base.
5. Generate Initial Table:
Uncheck it, if users don't need an initial table in the project output.
6. External Symbol Files:
Input the other symbol files (*.sym) needed for reference link in the current project.
7. Library modules:
Specify and show all library-modules included in the current project.

[Section]



1. Obj & Lib modules:

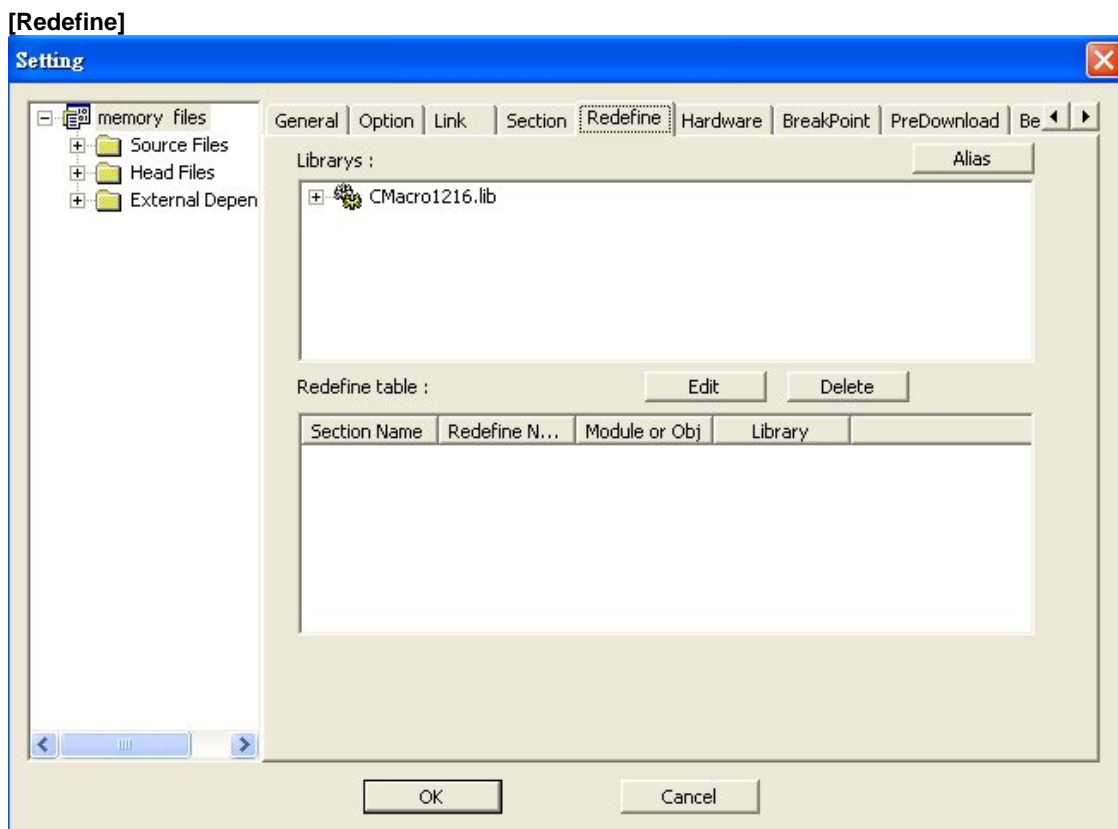
Show all object and library modules in current project.

2. Merged section:

List the merged segments in current project.

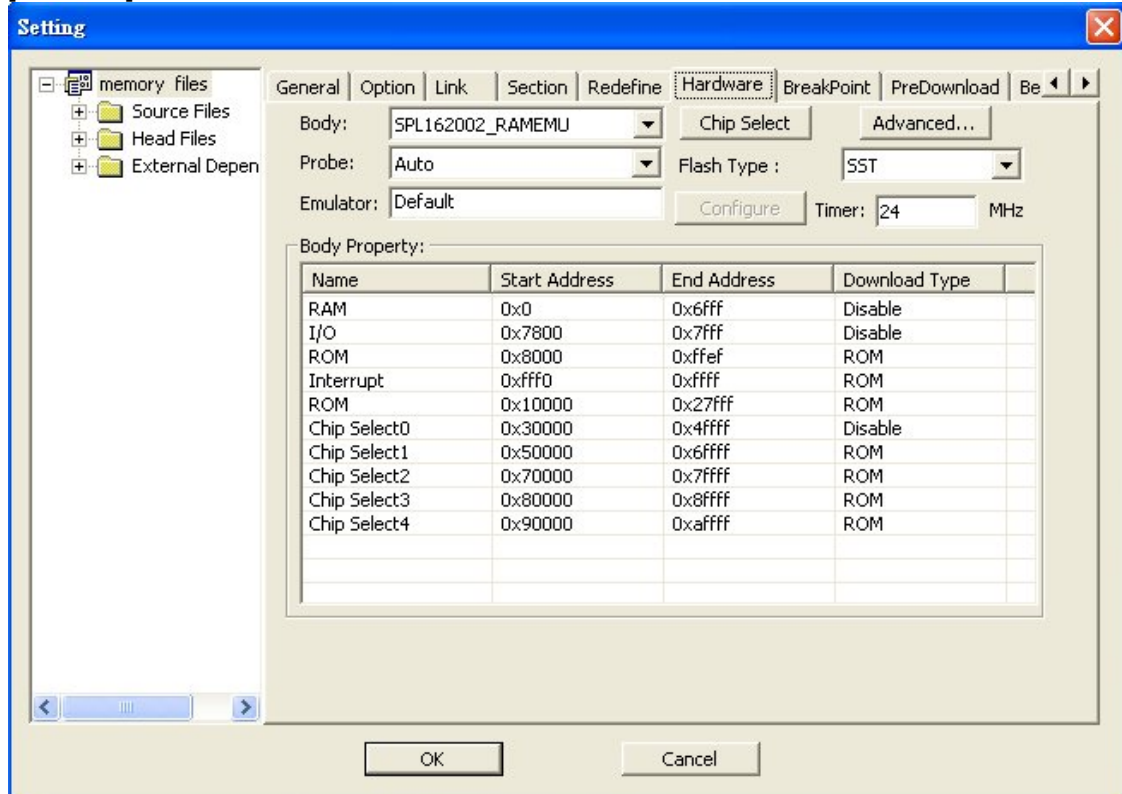
3. Non-merged section:

List the non-merged sections in current project. You can change the address or align base of these sections by double click on the ROM field, and the specified sections will be located at proper aligned addresses after re-linking this project.



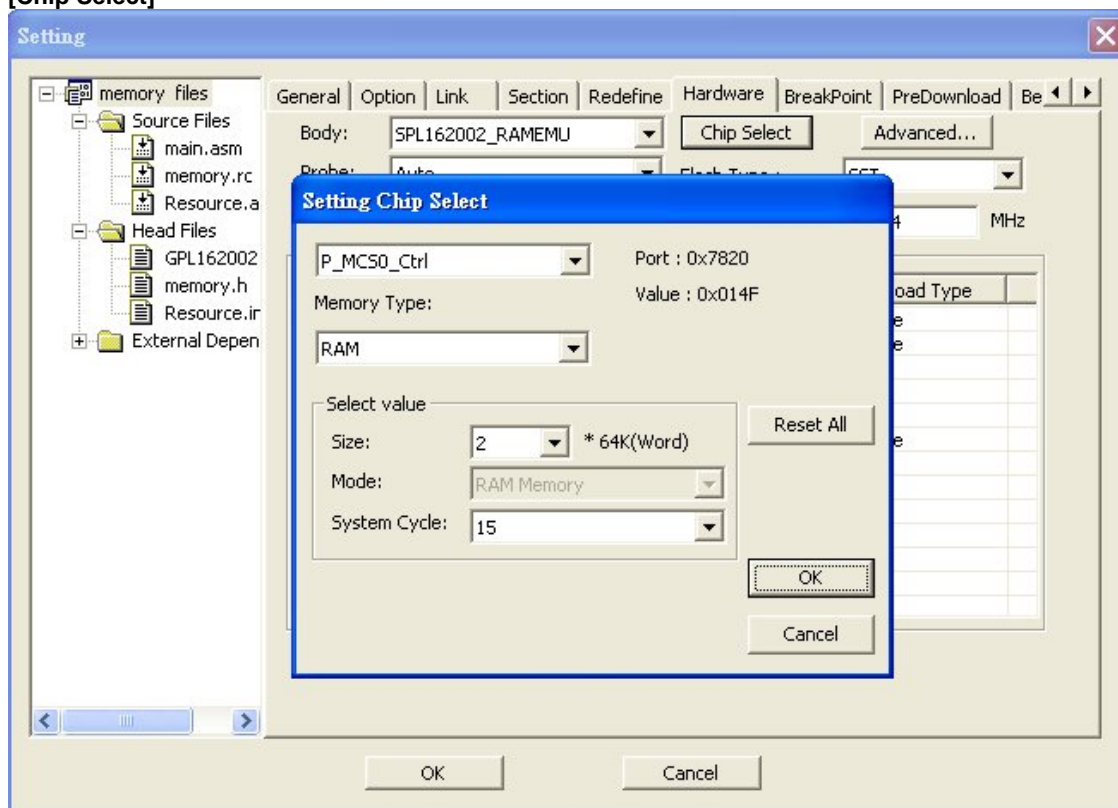
1. Alias: Select an section from the list window of libraries and rename it.
2. Edit: Edit the selected item in the Redefine table list window. It is the same with double clicking on the item in the Redefine table list.
3. Delete: Delete a selected item in the Redefine table list window.

[Hardware]



1. Body:
Select a body. The linker and simulator are based on the body description to link and simulate.
2. Emulator:
Select the external device emulator for selected IC, the emulator is a DLL specified at CPT file of a corresponding body.
3. Timer:
Set clock frequency of Simulator and Emulator.
4. Configure:
Set emulator.
5. Body property:
Show the memory mapping.

[Chip Select]



Chip Select is the setting of downloading a program and resources onto four external memories (CS0, CS1, CS2, CS3, and CS4) on EMU board through ICE. Programmers must set it up before downloading a program onto EMU board.

Note: This setting is only a reference for downloading a program from ICE to EMU board. Programmers must set Chip Select properly based on their needs.

1. Memory Chip Select:

Select which chip to be set up

2. Memory Type:

Which type of memory is used, e.g. RAM, ROM, Flash; in download box, check is disabled if this chip is not used. IDE, therefore, will not download a program into these memory devices while re-downloading. The disable sequence must be CS4, CS3, CS2, CS1 then CS0 can be disabled individually.

3. Select value:

a. Size: setup size of the chip.

The starting address of the GPL162002 CS# is determined by the previous memory size (CS0 is fixed in 0x30000). Therefore, the starting address of next memory must be appointed. The starting address of memory must be corresponding to the one given in the programming guide.

Some memory limitation may be applied: for example, to configure CS0→ 256K-word ROM, CS1 → 512K-word flash, CS2→ 512K-word flash, CS3 → 256K-word flash. To follow the limitation mentioned in the programming guide,

Set CS0 size to "4" (start address 0x30000)

Set CS1 size to "8" (start address 0x70000)

Set CS2 size to "8" (start address 0xF0000)

Set CS3 size to "4" (start address 0x130000)

b. Wait cycle:

The ICE downloading speed is 10MHz, i.e. CPU cycle = 100ns. If programmers use lower access-speed memory, it will have additional wait cycles to prevent download failure. In general, the extra wait cycle can be set as "0". This is because memory access time is seldom larger than 200ns (based on CPU cycle).

4. Internal Memory Setting:

Set the extra wait cycles for downloading into a memory device that tries to emulate internal mask ROM.